

STM32G4 Mixed-signal MCU workshop

Learn to design with the new STM32G4 mixed-signal microcontroller



Technology Tour 2019

Vancouver, BC | September 24





STM32G4 mixed-signal MCU workshop

2

Agenda

1:00PM – 5:00PM

Introduction to the STM32G4 Series
Hands On: STM32G474 Discovery Kit Demo
Hands On: ADC Gain Compensation
Hands On: DAC Signal Generation
Hands On: Programmable Gain Amplifier
Hands On: High Resolution Timer (HRTIM)
Hands On: CORDIC Hardware Accelerator
Hands On: Filter Math Accelerator (FMAC)
Hands On: Dual Bank Flash Live Firmware Upgrade
Hands On: CCM-SRAM
Wrap Up

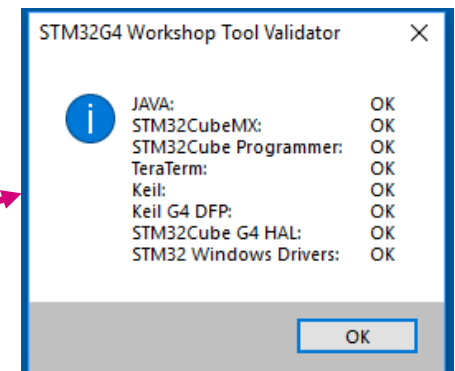


All set for the Hands-On?

Per your email invitation, you should have installed on your System:

- **STM32CubeMX:** v5.3.0 minimum
- **STM32CubeProgrammer:** v2.1.0 minimum
If you installed **v2.1.0**, also execute
"C:\Program Files (x86)\STMicroelectronics\STM32Cube\STM32CubeProgrammer\Drivers\stsw-link009_v3\dpinst_amd64.exe"
- **STM32CubeG4:** v1.1.0 minimum
- **KEIL MDK-ARM:** v5.28a minimum
- **KEIL MDK-ARM LICENSE:** Pro Eval License Add On
- **Tera Term** or similar

*Run the validator
to check status*



Please ask for help if you have any problems

STM32 General Usage Survey

Please complete this survey prior to the start of the workshop

Use your phone to scan the QR code or type the link into your browser.



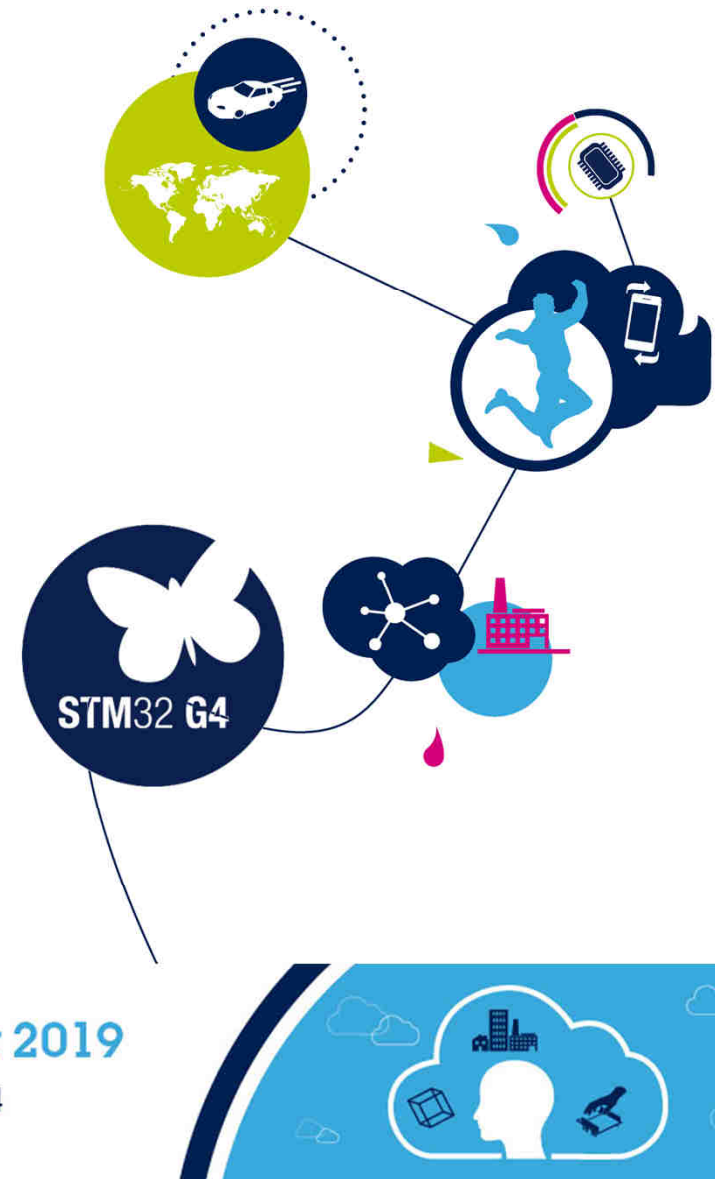
<https://www.surveymonkey.com/r/68NZNQR>



We Greatly Value Your Feedback - Thank you!

STM32G4 Mixed-signal MCU workshop

*Learn to design with the new STM32G4
mixed-signal microcontroller*



Technology Tour 2019

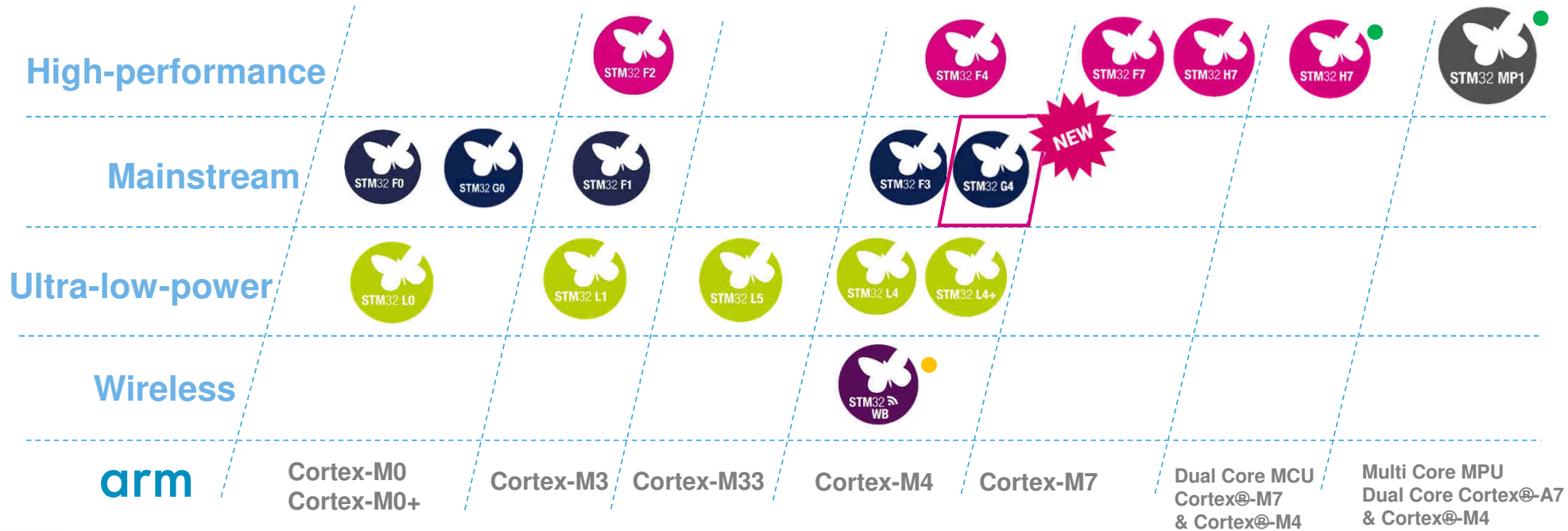
Vancouver, BC | September 24



STM32 portfolio

6

17 product series / More than 50 product lines



Note ● : Cortex-M0+ Radio Co-processor, ● : Cortex-M4 Co-processor

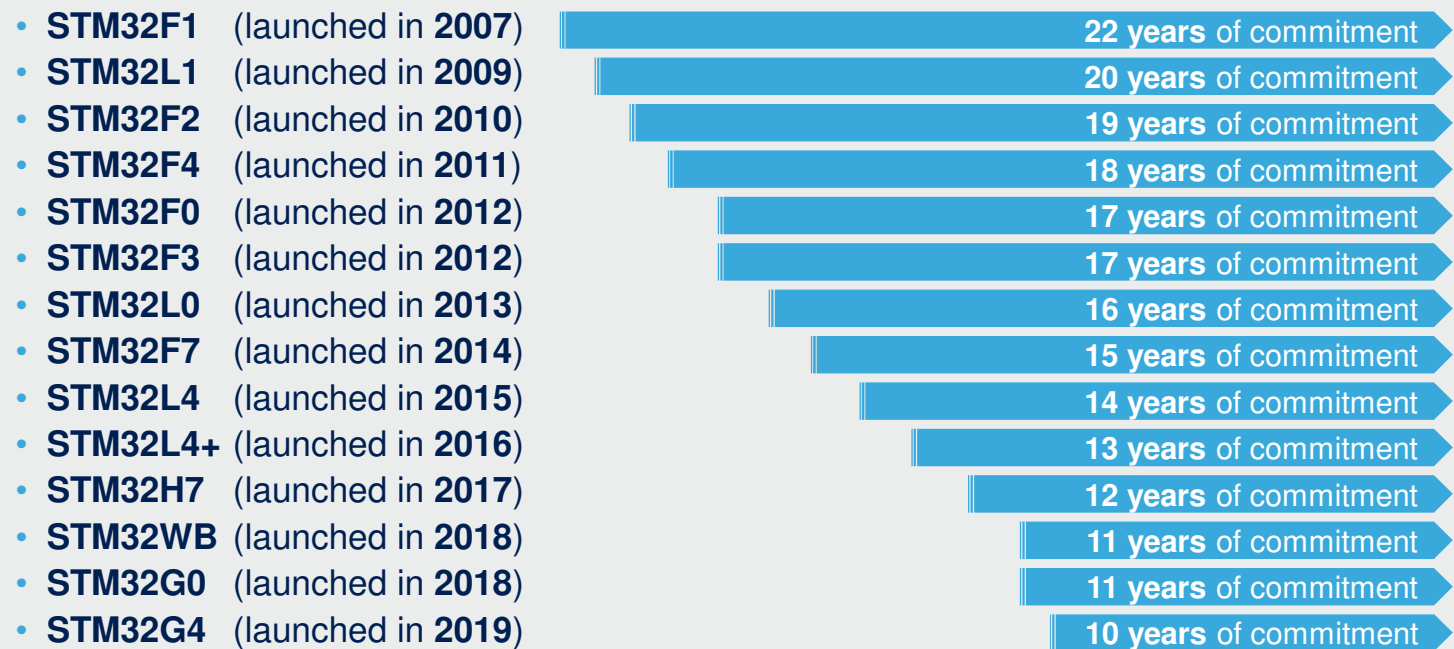
STM32 Rolling Longevity Commitment

7

Longevity commitment is renewed every year



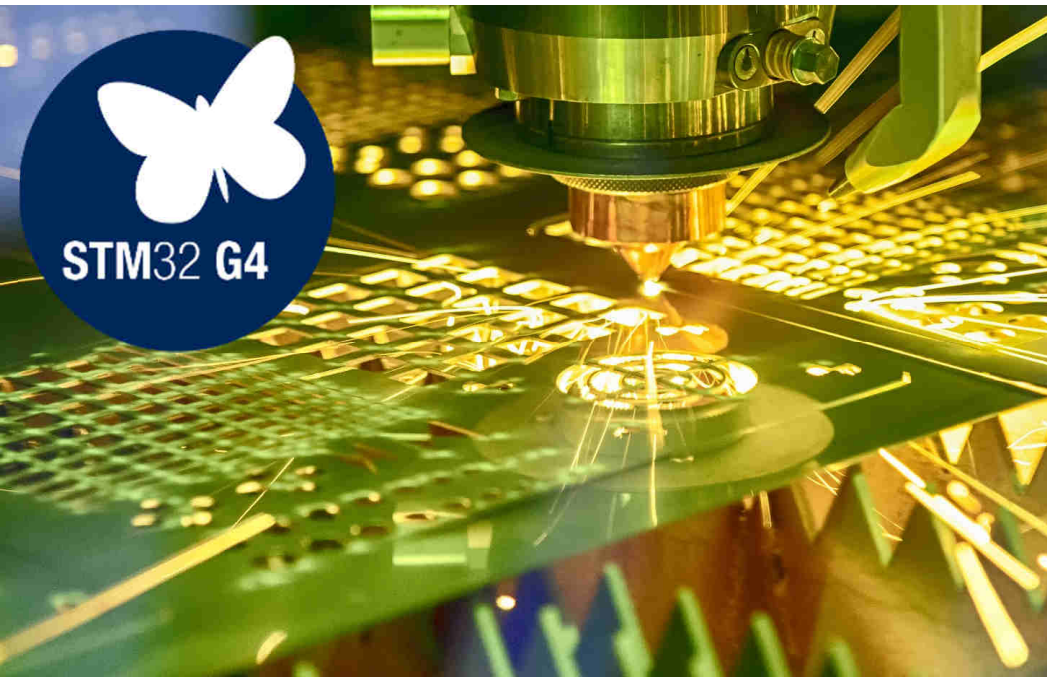
starting January
1st 2019
→ Until 2029



STM32G4 Series

8

Ideal for applications requiring MCU with advanced and rich analog peripherals



- Control applications (Motor Control...)
- Industrial equipment
- Instrumentation and Measurement
- Digital Power
 - Digital SMPS (switch mode power supply)
 - PFC (power factor correction)



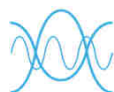
STM32G4 Series – Key Messages

9



Performance

- Arm® Cortex®-M4 at 170 MHz
- 213 DMIPS and 550 CoreMark® results
- Better dynamic power consumption (163µA/MHz)
- ART Accelerator™ (dynamic cache)
- Mathematical accelerators
- CCM-SRAM Routine Booster (static cache)



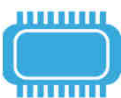
Rich Integrated Analog and Digital

- Op-Amps (Built-in gain), DACs, Comparators
- 12-bit ADCs 4Msps with hardware oversampling
- CAN-FD (flexible data rate – 8Msps bit rate)
- High resolution timer (184 ps)
- USB type-C Power Delivery3.0
- 1% RC accuracy [-5°..90°C], 2% full T° range



Safety and security focus

- Dual Bank Flash with ECC (error code correction)
 - Securable Memory Area
 - Hardware encryption AES-256
 - SIL, Class-B
 - SRAM with Parity bit
- } Secure Live Upgrade
- } Functional safety design packages



Complete portfolio

- Complements existing STM32F3 Series portfolio
- From -40°C up to 85 or 125°C devices
- From 32- up to 128-pin
- From 32KB to 512KB Flash

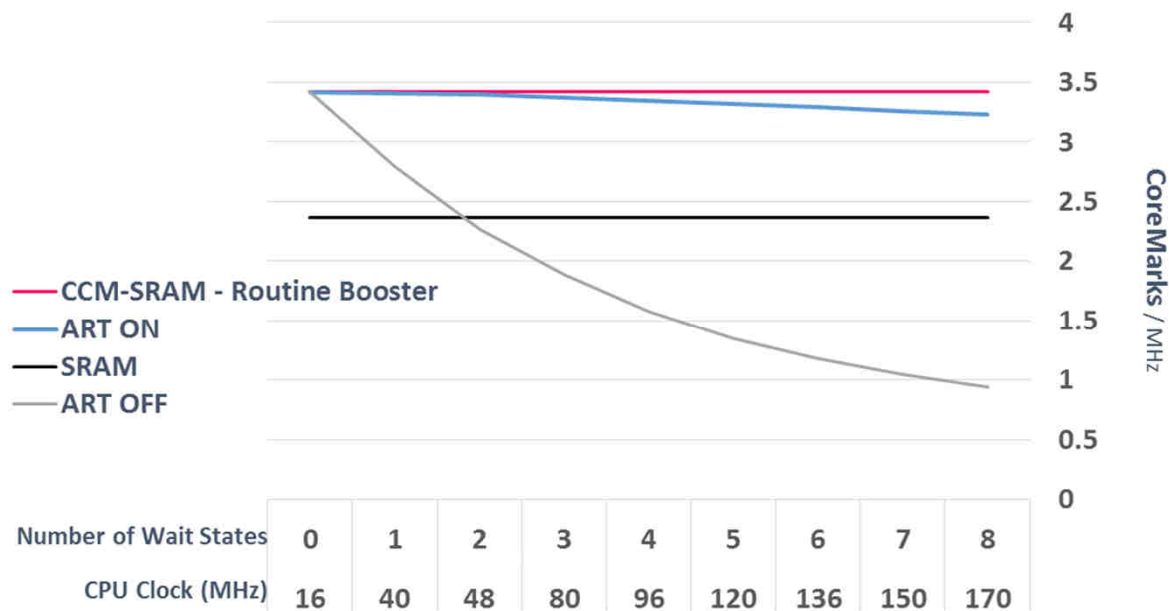


Greater Performance

10

Pure 170 MHz CPU performance (Arm® Cortex®-M4) with 3 accelerators

Code execution performance



Arm Cortex-M4 with **FPU**

Up to 170 MHz CPU frequency

Up to 213 DMIPS and 550 CoreMark® results

3 different HW accelerators:

- **ART accelerator** (~dynamic cache)
→ Full code acceleration (average)
- **Routine Booster CCM-SRAM**
(~static cache) → determinism preserved
- **Mathematical** (CORDIC + FMAC)





Mathematical Accelerators

11

Function acceleration and CPU offload

1. CORDIC (Trigo)

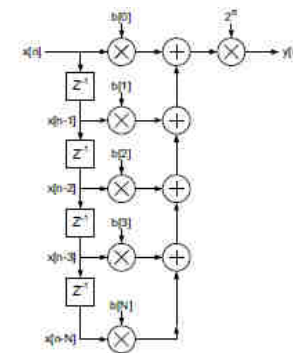
- Very helpful for Field Oriented Motor Control method (FOC)

- Vector rotation (polar to rectangular): Sin, Cos
- Vector translation (rectangular to polar): Atan2, Modulus
- Sinh, Cosh, Exp
- Atan, Atanh
- Square root
- Ln

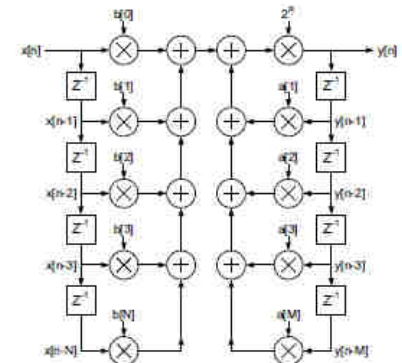
2. Filter Math Accelerator (FMAC)

- Can be used to create
 - 3p3z Compensator ($\rightarrow$ Digital power)
 - Sigma Delta modulator
 - Noise Shaper

FIR filter



IIR filter

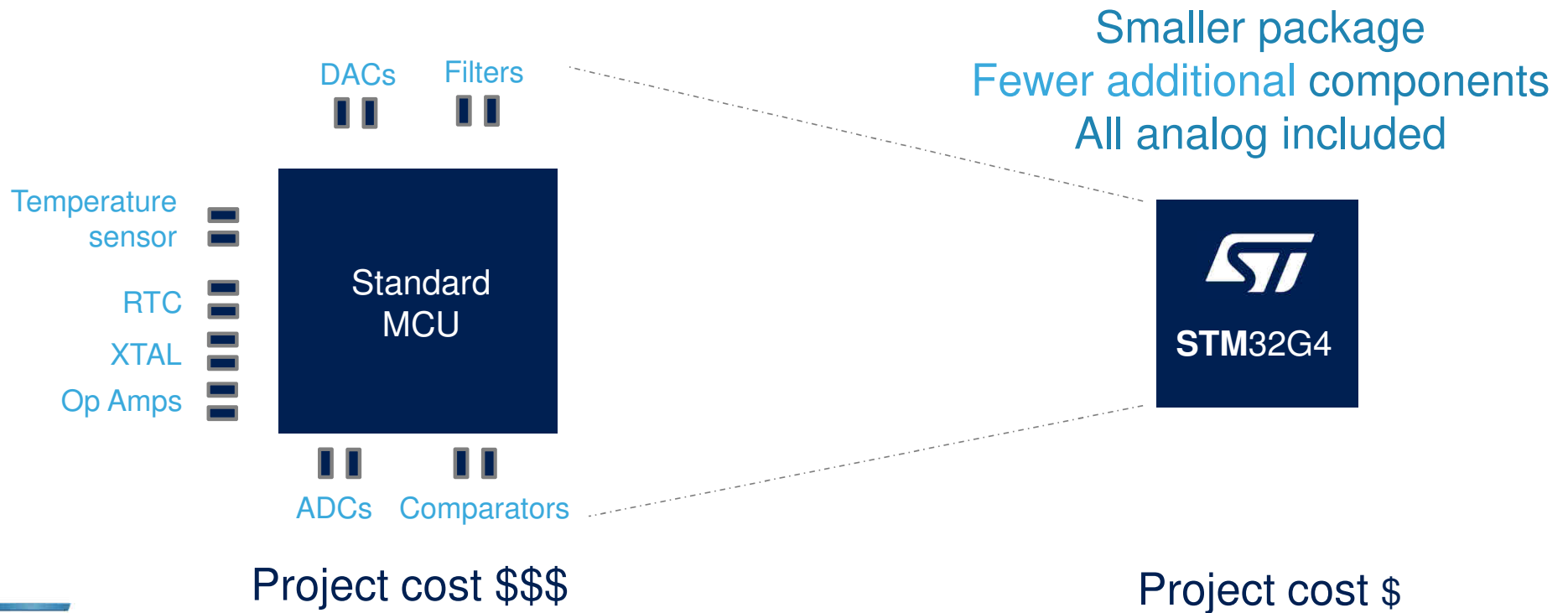




Reducing PCB Size and BOM Cost

12

System-on-Chip – All-in-one solution





Rich, Advanced Analog

13

Mixed-signal SoC for wide variety of applications

ADC (up to 5)	Values
Topology	SAR 12-bit + HW oversampling → 16-bit
Sampling rate	Up to 4 Msps
Input	Single-ended and differential
Offset and Gain compensation	Auto calibration to reduce gain and offset

DAC (up to 7)	Values
Sampling rate	15 Msps (internal) 1Msps (from buffered output)
Settling time	16ns

Op-Amp (up to 6)	Values
GBW	13 MHz
Slew rate	45 V/μs
Offset	3mV over full T° range 1.5mV @ 25°C
PGA Gain (accuracy)	2, 4, 8, 16, -1,-3,-7,-15 (1%) 32, 64, -31,-63 (2%)

Comparator (up to 7)	Values
Power supply	1.62 .. 3.6V
Propagation delay	16.7ns
Offset	-6 .. +2 mV
Hysteresis	8 steps: 0, 9, 18, 27, 36, 45, 54, 63 mV



Key Features for Targeted Applications

14

Motor Control

Home appliances, E-bikes, Air Conditioning

- Fast CPU 170MHz
- Mathematical accelerator (Cordic)
- Advanced Motor Control timers
- Fast comparators
- 4Msps ADC-12bit + HW oversampling
- Op-Amp with built-in gain (PGA)
- DAC-12bit
- 1% RC accuracy (UART communication w/o external Xtal)



High-End Consumer

Rechargeable devices, drones, toys

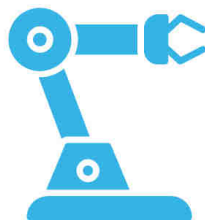
- Low-thickness, small form-factor
- Low consumption in run mode ~ 160µA/MHz
- Embedded analog
- SAI (Sound Audio Interface)
- USB type-C Power Delivery 3.0



Industrial devices Measurements

Industrial equipment

- Fast CPU 170MHz
- Mathematical accelerator (Cordic)
- High temperature 125°C
- CAN FD support
- SPI, USART, I²C
- Advanced timers
- Real Time Clock with backup registers
- Dual bank flash for **live** upgrade
- AES & security



Digital Power

Servers, Telecom, EV Charging station

- Fast CPU 170 MHz
- Mathematical accelerator (Filtering)
- 12ch High Resolution timer (184ps)
- 4Msps ADC-12bit + HW oversampling
- Fast comparators (17ns)
- Embedded analog
- Dual bank flash for **live** upgrade
- AES & security
- FMAC for 3p3z compensation

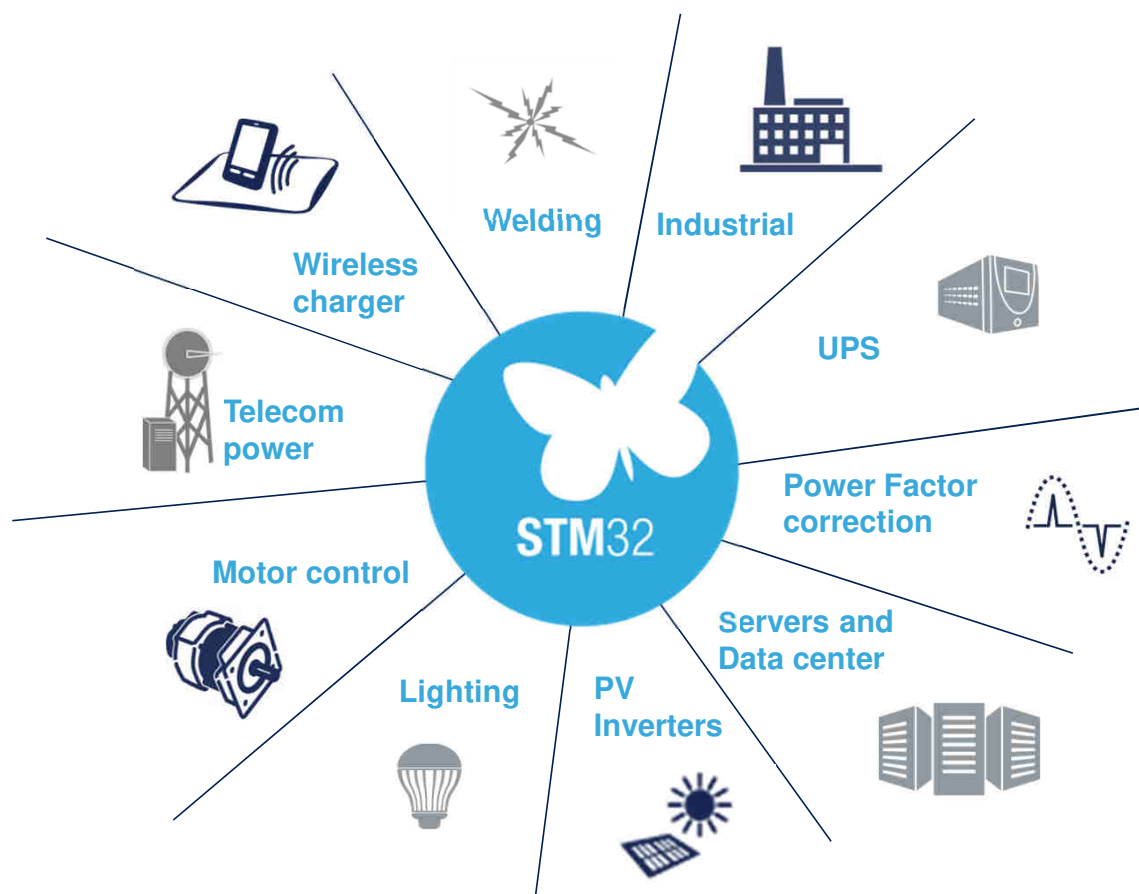




Ease Digital Power Conversion

15

Enhance your digital power solutions using the STM32G4's full features High Resolution Timer (HRTIM)





HRTimer – Not only High Resolution...

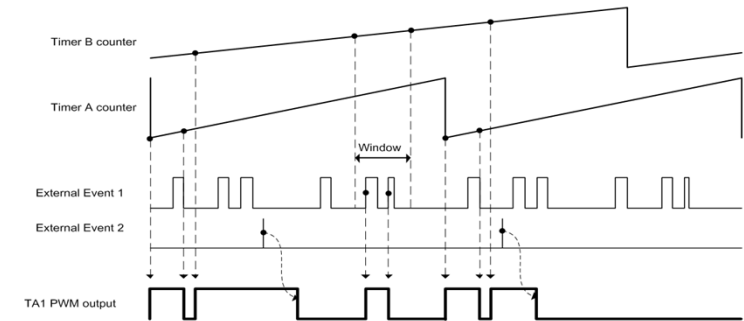
16

High resolution PWM

- 12 channels with 184ps resolution on frequency and duty cycle
- 184ps is equivalent to 5.4GHz timer clock

Flexible PWM generation

- 7x independent time base to create various shape of PWM
- 6x complementary pair PWM outputs
- Up to 32 set/reset transition per PWM period thx to the built-in crossbar
- Master/Slave configuration for multi phase converter



Multiple Event handler

- 6x Digital and Analog fault input
- 10x Events cycle to cycle current control or PWM restart (constant Ton/Toff)
- Blanking, windowing and digital filter

12 independent channels

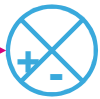
- Any topology supported from 1x 12 PWM (triple interleaved LLC (servers application) up to 12x1 PWM (multiple independent buck converters (lighting))



Shaped for Control

17

Set point



ARM Cortex-M4 core @ 170MHz

- FPU
 - Enhance dynamics
 - No scaling overhead
 - No saturation
- DSP (fast MAC)
- SIMD
- Parallel processing
- Low interrupt latency

ST's product architecture

- ART accelerator
- Wait state removal
- CCM-SRAM accelerator
- Real time execution
- Math accelerator
 - Cordic (Trigo)
 - FMAC (Filtering)

5x 12-bit 4Msps ADC

- SAR (no pipeline delay)
- Low latency (250ns)
- Low aperture time (20ns) for snapshot measurements
- Simultaneous sampling on multiple ADCs
- HW oversampling

PWM Timers

- * 170 MHz (5.9ns)
- * HRTIM (184ps)

PWM

PLANT

Direct HW path (no latency)

- Instantaneous control load
- Protection

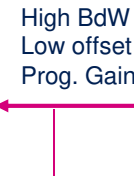


7xDAC

Multiple fast Comparators

Analog feedbacks

Digital feedback



High BdW
Low offset
Prog. Gain

Other Timers

- Quad encoder
- Halls sensors

Easy to use the Analog and Digital resources thanks to high peripherals interconnect and flexible bus matrix



Greater Security

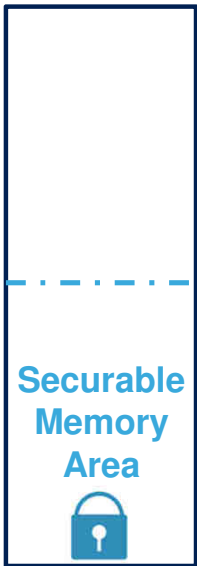
18

Integrated security features, ready for tomorrow's needs

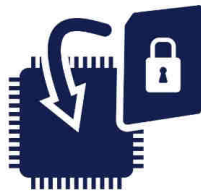
User Flash

Bank1

Bank2



STM32Trust: a complete toolset
an ecosystem for embedded security



<https://www.st.com/stm32trust>



Securable Memory Area:

- Configurable size
- Can be secured once exiting
- No more access nor debug possible
- Good fit to store critical data
- Critical routines
- Keys



Securable user memory AES TRNG PCROP MPU Readout protection CRC Write Protection

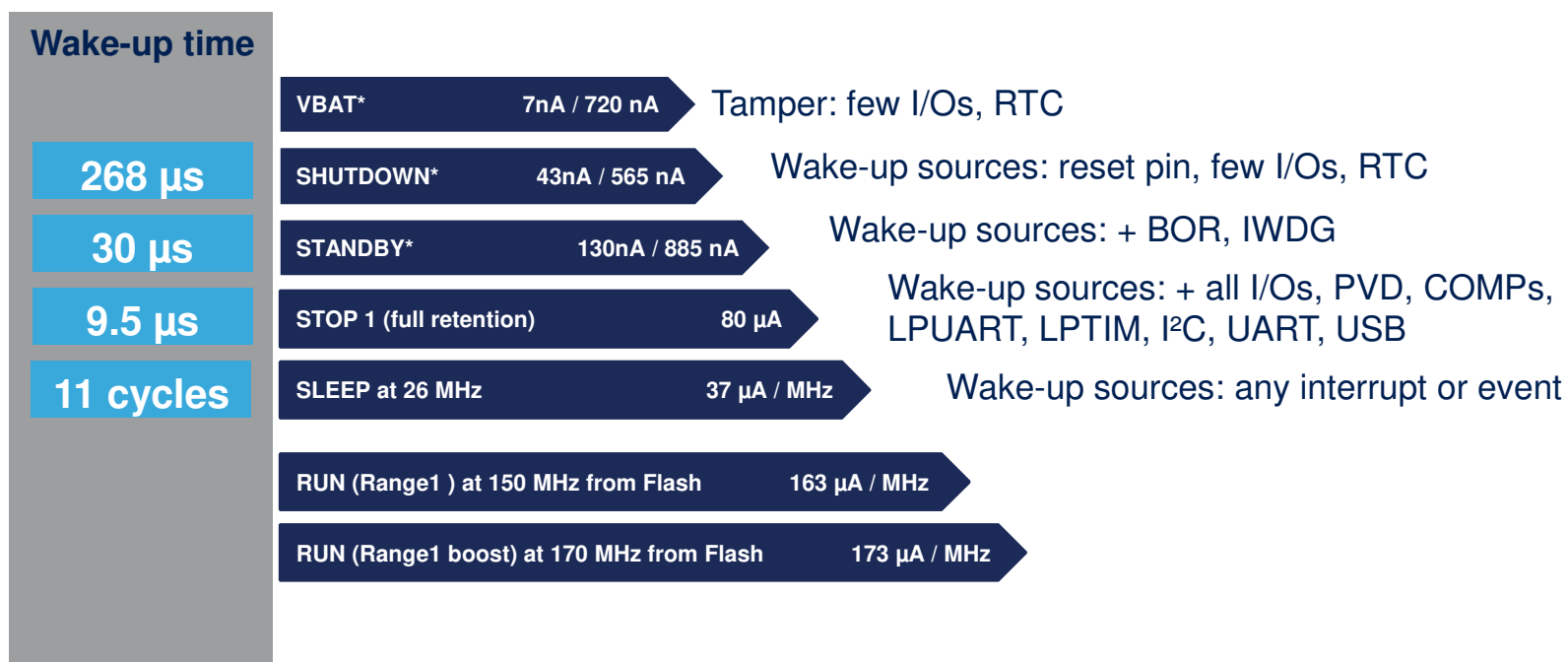
	Securable user memory	AES TRNG	PCROP	MPU	Readout protection	CRC	Write Protection
Secure firmware install (SFI)	●	●			●		
Secure Firmware upgrade (SFU)	●	●			●		●
Mutual Distrustful			●				
Firmware IP protection			●				
Secret key storage	●				●		
Secured communication		●				●	
Authentication	●	●			●		
Task cloisoning				●			



Dynamic Efficiency Modes

19

When Mainstream MCU Series meets low-power requirements



Conditions: 25°C, $V_{DD} = 3V$

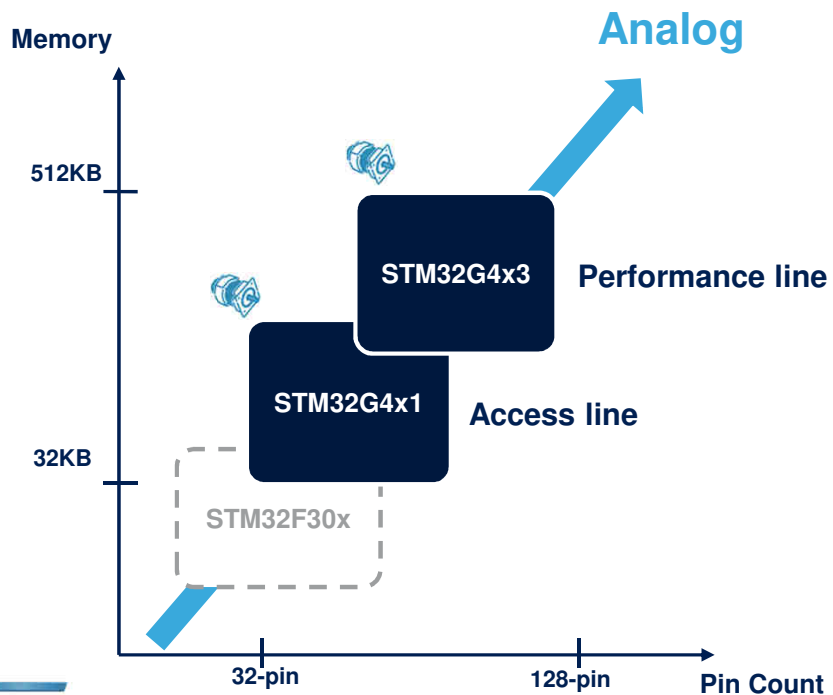
Note : * without RTC / with RTC



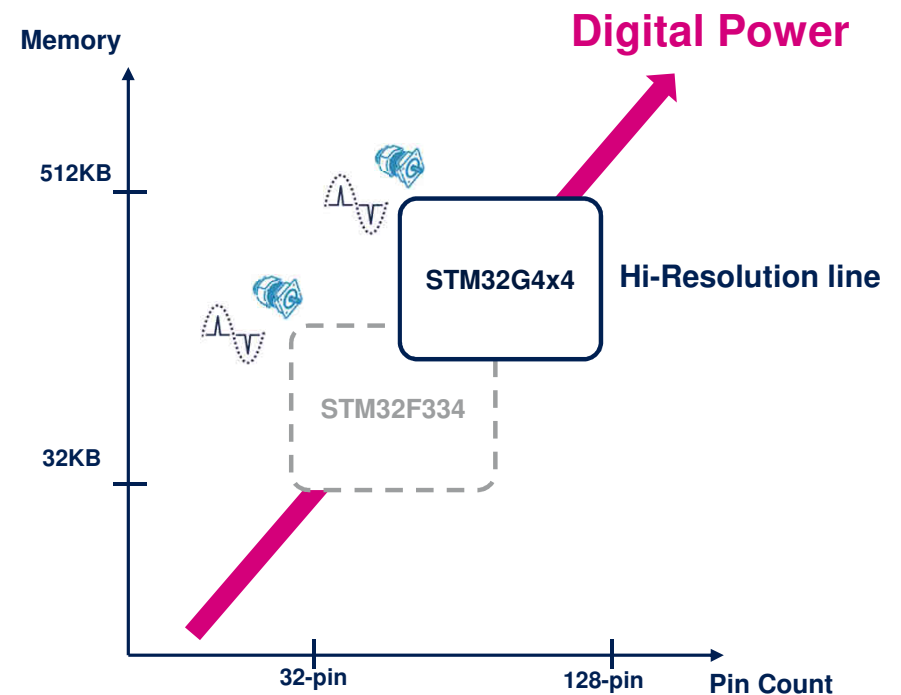
STM32G4 Products Lines

20

General Purpose



Applications Specific





Extensive & Innovative Peripheral Set

21

No compromise on what matters

Unit parameters	STM32G474 Hi-Resolution line	STM32G473 Performance line	STM32G431 Access line
Core, frequency	Arm Cortex-M4, 170 MHz		
Flash (max)	512 Kbytes (2x256KB dual bank)		128 Kbytes single bank
RAM (up to)	96 Kbytes		22 Kbytes
CCM –SRAM (code-SRAM)	32 Kbytes		10 Kbytes
12-bit ADC SAR	4x 12-bit 4 MSPS		2x 12-bit 4 MSPS
Comparator	7		4
Op amp with 4 built-in gain values with 1% accuracy	6		3
12-bit DAC	7		4
Motor Control timer	3x (170 MHz)		2x (170 MHz)
CAN-FD	3x		1x
12 channel Hi-resolution Timer	1x	-	-
Power supply	1.72 to 3.6 V		



High Resolution and Performance lines [128KB .. 512KB]

- **32-bit Arm Cortex-M4 core with FPU**
- **ART + CCM-SRAM + Mathematic Accelerators**
- **Dual Bank Flash with ECC**
- **SRAM with Parity bit**
- **+/- 1% internal clock**
- **1.72 to 3.6V power supply**
- **Up to 125°C**

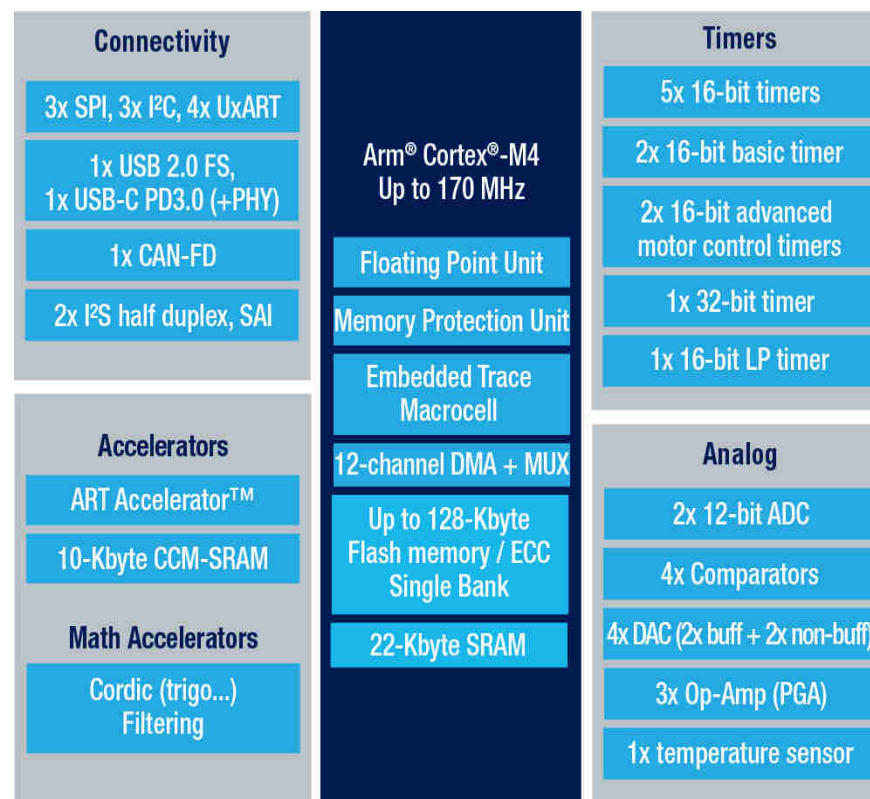


- **High resolution timer**
- **3x Advanced Motor Control timers**
- **Rich Advanced Analog**
- **3x CAN Flexible Data rate**
- **USB-C Power Delivery3.0**
- **Advanced Security and Safety features**
- **Robustness: highest level 5 / FTB/ESD - IEC 61000-4-4**



Access line [32KB .. 128KB] and up to 512KB in H1-2020 !

- **32-bit Arm Cortex-M4 core with FPU**
- **ART + CCM-SRAM + Mathematic Accelerators**
- **Single Bank Flash with ECC**
- **SRAM with Parity bit**
- **+/- 1% internal clock**
- **1.72 to 3.6V power supply**
- **Up to 125°C**



- **2x Advanced Motor Control timers**
- **Rich Advanced Analog**
- **CAN Flexible Data rate**
- **USB-C Power Delivery3.0**
- **Advanced Security and Safety features**
- **Robustness: highest level 5 / FTB/ESD - IEC 61000-4-4**



STM32G4 Portfolio

24



Legend: Crypto AES-256 Available in H1 2020



Broad Portfolio

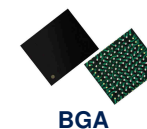
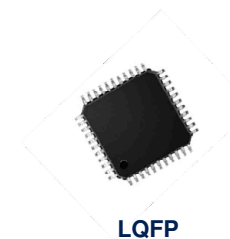
25

Portfolio extended to support budget applications efficiently

More memory and pin counts

Flash memory (bytes)	32-pin LQFP QFN	48-pin LQFP QFN	64-pin LQFP BGA WLCSP	80-pin LQFP WLCSP	100-pin LQFP BGA	121-pin BGA	128-pin LQFP
512 K		✓	✓	✓	✓	✓	✓
256 K		✓	✓	✓	✓	✓	✓
128 K	✓	✓	✓	✓	✓	✓	✓
64 K	✓	✓	✓	✓	✓		
32 K	✓	✓	✓	✓	✓		

More packages



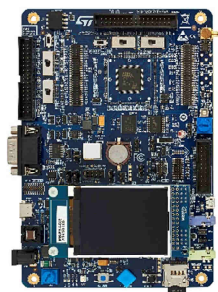
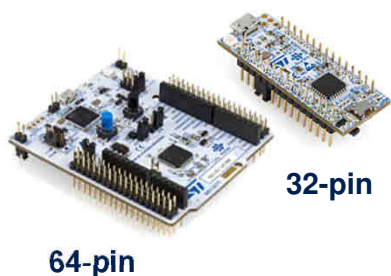
 Note: new packages in STM32 portfolio



STM32G4 Hardware Solutions

26

Accelerate evaluation, prototyping and design



STM32 Nucleo

Flexible prototyping

- NUCLEO-G431RB
- NUCLEO-G474RE
- NUCLEO-G431KB

Evaluation boards

Full feature STM32G4 evaluation

- STM32G484E-EVAL
- STM32G474E-EVAL
- STM32G474E-EVAL1

Motor Control Pack

Full feature for Motor Control and Analog

- P-NUCLEO-IHM03

Discovery kits

Key feature prototyping

- B-G474E-DPOW1
- B-G431B-ESC1

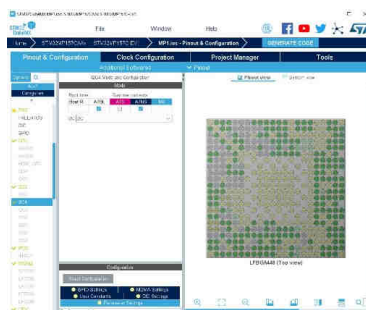


STM32G4 Software Tools

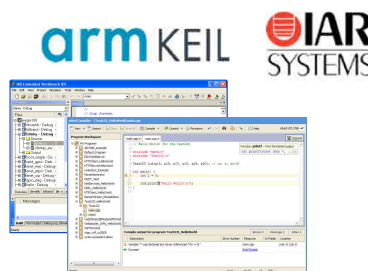
27

Complete support of Arm Cortex-M ecosystem

STM32
CubeMX



eclipse
STM32
CubeIDE



All-in-one STM32 programming tool
Multi-mode, user-friendly



STM32CubeMX

STM32CubeMX

- Configure and generate Code
- Conflicts solver

IDEs Compile and Debug

Flexible Solutions

- Partners IDE, like IAR and Keil
- Free IDE based on Eclipse, like STM32CubeIDE*

STM32 Programming Tool

STM32CubeProgrammer

- Flash and/or system memory
- GUI or command line interface

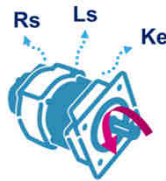


Dedicated Ecosystems

28

Motor Control

- **Complete ecosystem** (HW boards, SW Development Kit (SDK), docs and trainings)
 - **X-CUBE-MCSDK** (v5.4)
 - Motor Control FW library based on STM32Cube HAL and LL
 - Motor control workbench: Graphical configurator of the motor control library linked with STM32CubeMx
 - **P-NUCLEO-IHM03**: Motor Control Nucleo pack
 - NUCLEO-G431RB Nucleo-64
 - X-NUCLEO-IHM16M1 motor driver expansion board
 - Low Voltage motor
- **State of the art algorithms** (FOC, 6-step, sensorless...)
- **Motor Profiler**: Plug and spin your motor within less than one minute



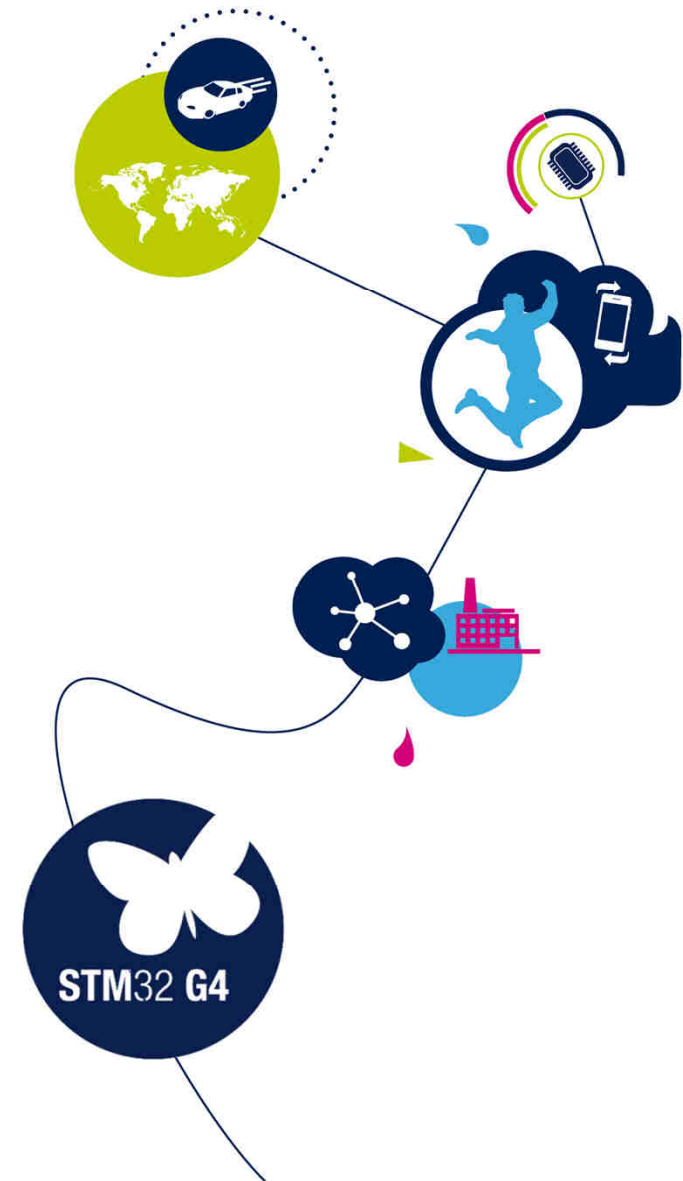
Digital Power

- **Complete ecosystem** (HW boards, FW examples, SW tools, docs and trainings)
- **Dedicated HRTIM Cook Book - AN4539**: How to operate the Hi-Resolution timer in different topology
- **Digital Power training** (PSU and PFC) – based on STM32 G4 series – done in collaboration with Biricha (from Q4 2019)



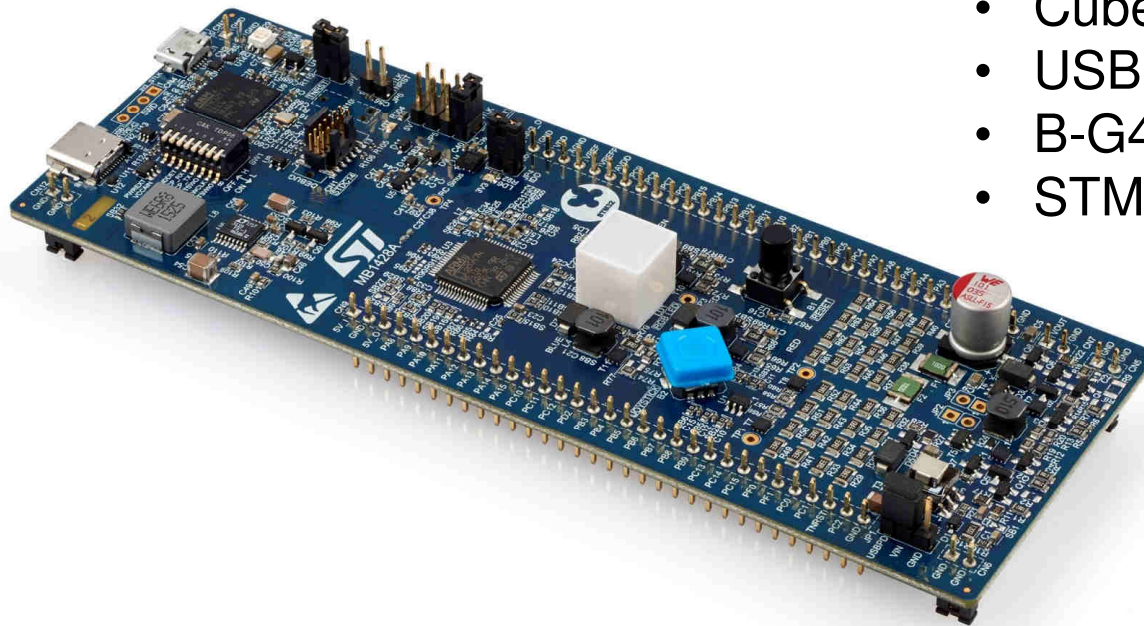
STM32G4 mixed-signal MCU workshop

*Hands On:
STM32G4 Discovery Kit Demo*





Systems Check



- Cube Programmer
- USB cable
- B-G474E-DPOW1 Discovery Board
- STM32G4 Family Cube Library



B-G474E-DPOW1

31

Jumpers – 1 of 2

**Do not attach
USB cable here**

**All switches
Down**

CN2
USB Type-C™
receptacle

SW1
Octal mechanical
switch

U6
LTC3114-1

CN1
STDC14 debug
connector

LD7
5V LED

LD6
3V3 LED

CN3
USB micro-B
ST-LINK receptacle

LD9
COM LED

LD8
overcurrent LED

U9
STLINK-V3E MCU

JP7
TNRST

JP6
STLK NRST

JP5
5V_SELECTION

**Attach USB
cable here**

INSTALLED

OPEN

INSTALLED on STLK

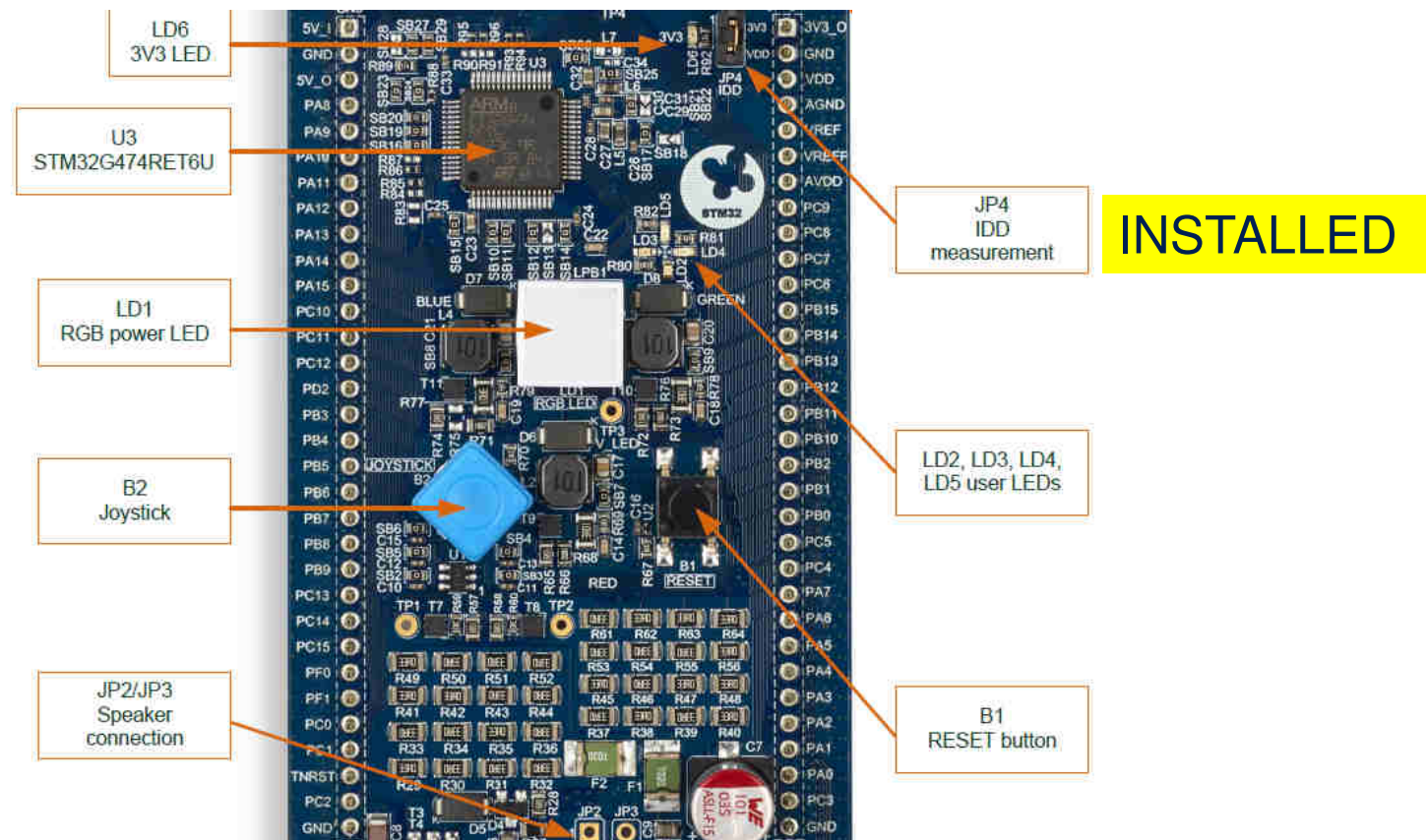




B-G474E-DPOW1

32

Jumpers – 2 of 2

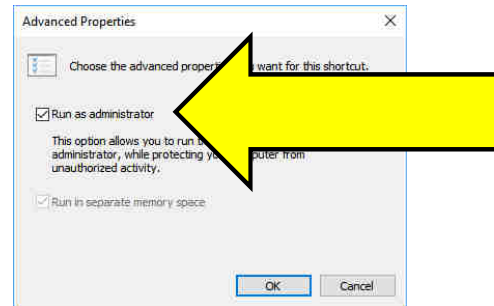
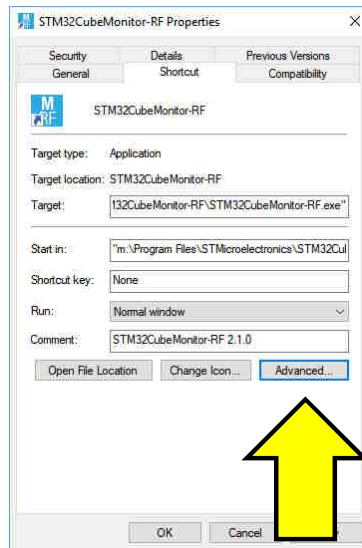




CubeProgrammer

33

Start up

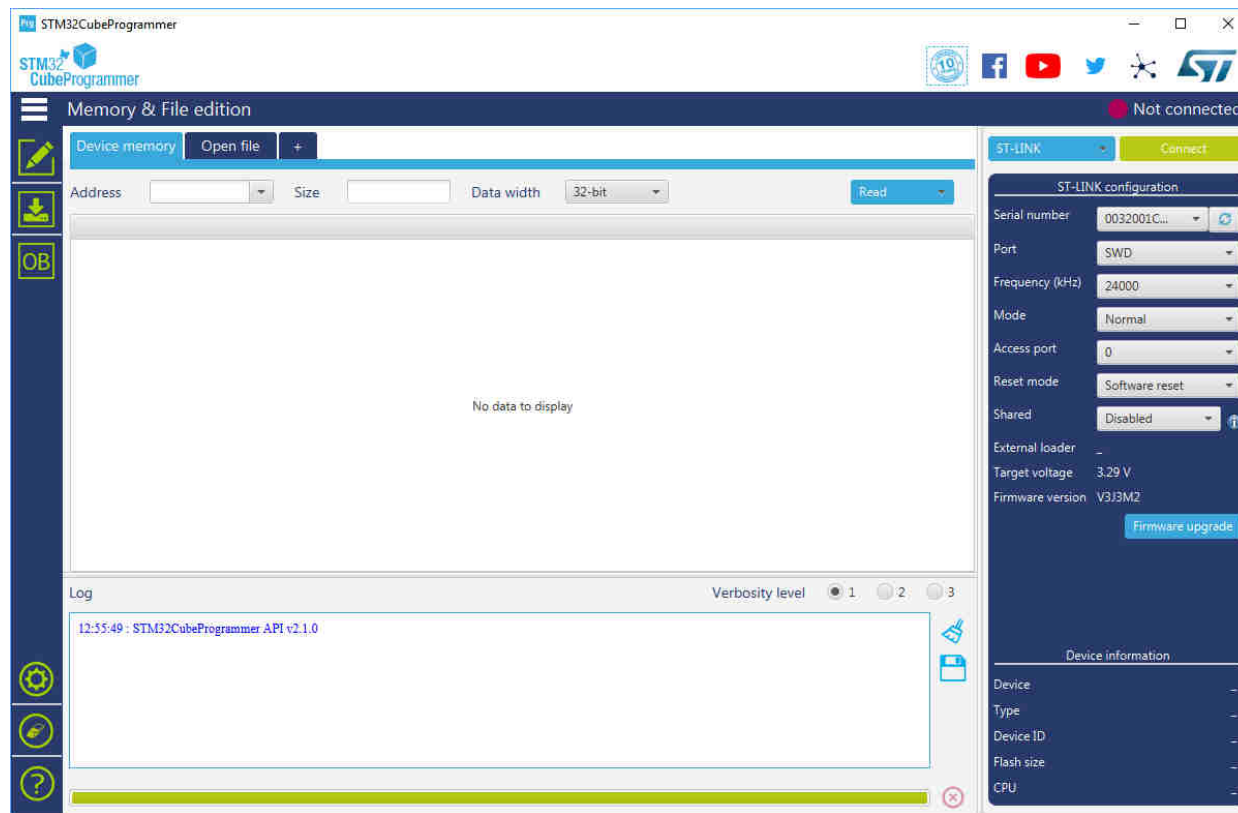




CubeProgrammer

34

Splash Page - Connect



Connect



CubeProgrammer

35

Splash Page – Open File

The screenshot displays the STM32CubeProgrammer software interface. A yellow arrow points to the 'Open file' button in the 'Memory & File edition' section. The interface includes a table for memory data, a log window, and a right-hand panel for ST-LINK configuration and device information.

Memory & File edition

Device memory | Open file

Address: 0x08000000 Size: 16 Bytes

Address	0	4	8	C	ASCII
0x08000000	20000560	080001ED	08002D6D	08002A31	...i...m...1*...
0x08000010	08002D69	0800052D	08003009	00000000	i-...0.....
0x08000020	00000000	00000000	00000000	08002DED	i-..
0x08000030	080005F3	00000000	08002D6F	08002DEF	6.....0-..i-..
0x08000040	08000207	08000207	08000207	08000207	
0x08000050	08000207	08000207	08000207	08000207	
0x08000060	08000207	08000207	08000207	08000531	1...
0x08000070	08000207	08000207	08000207	08000207	
0x08000080	08000207	08000207	08000207	08000207	
0x08000090	08000207	08000207	08000207	08000207	
0x080000A0	08000207	08000207	08000207	08000207	
0x080000B0	08000207	08000207	08000207	08000207	
0x080000C0	08000207	08000207	08000207	08000207	

Log

Verbosity level: 1 2 3

12:58:46 : Size: 16 Bytes
12:58:46 : Bank: 0x02
12:58:46 : Address: 0x0022070
12:58:46 : Size: 8 Bytes
12:58:46 : UPLOADING...
12:58:46 : Size: 1024 Bytes
12:58:46 : Address: 0x8000000
12:58:46 : Read progress:
12:58:46 : Data read successfully
12:58:46 : Time elapsed during the read operation is: 00:00:00.001

ST-LINK configuration

Serial number: 00000000
Port: COM1
Frequency (kHz): 24000
Mode: Normal
Access port: 0
Reset mode: Software reset
Shared: Disabled
External loader: ...
Target voltage: 3.29 V
Firmware version: V3.2M2
Firmware upgrade

Device information

Device: STM32G47x/G48x
Type: MCU
Device ID: 0x469
Flash size: 512 KB
CPU: Cortex-M4



CubeProgrammer

36

Find Demo.hex

Open File

2019_STM32G4_Workshop\HandsOn_Labs\Demo

Address	0	8	C	ASCII
0x08000000	00000000	00000000	00000000	00000000
0x08000008	00000000	00000000	00000000	00000000
0x08000010	00000000	00000000	00000000	00000000
0x08000018	00000000	00000000	00000000	00000000
0x08000020	00000000	00000000	00000000	00000000
0x08000028	00000000	00000000	00000000	00000000
0x08000030	00000000	00000000	00000000	00000000
0x08000038	00000000	00000000	00000000	00000000
0x08000040	00000000	00000000	00000000	00000000
0x08000048	00000000	00000000	00000000	00000000
0x08000050	00000000	00000000	00000000	00000000
0x08000058	00000000	00000000	00000000	00000000
0x08000060	00000000	00000000	00000000	00000000
0x08000068	00000000	00000000	00000000	00000000
0x08000070	00000000	00000000	00000000	00000000
0x08000078	00000000	00000000	00000000	00000000
0x08000080	00000000	00000000	00000000	00000000
0x08000088	00000000	00000000	00000000	00000000
0x08000090	00000000	00000000	00000000	00000000
0x08000098	00000000	00000000	00000000	00000000
0x080000A0	00000000	00000000	00000000	00000000
0x080000A8	00000000	00000000	00000000	00000000
0x080000B0	00000000	00000000	00000000	00000000
0x080000B8	00000000	00000000	00000000	00000000
0x080000C0	00000000	00000000	00000000	00000000
0x080000C8	00000000	00000000	00000000	00000000
0x080000D0	00000000	00000000	00000000	00000000
0x080000D8	00000000	00000000	00000000	00000000
0x080000E0	00000000	00000000	00000000	00000000
0x080000E8	00000000	00000000	00000000	00000000
0x080000F0	00000000	00000000	00000000	00000000
0x080000F8	00000000	00000000	00000000	00000000

Log

12:58:46 : Size: 16 Bytes
12:58:46 : Bank: 0x02
12:58:46 : Address: 0x40022070
12:58:46 : Size: 8 Bytes
12:58:46 : UPLOADING...
12:58:46 : Size: 1024 Bytes
12:58:46 : Address: 0x8000000
12:58:46 : Read progress:
12:58:46 : Data read successfully
12:58:46 : Time elapsed during the read operation is: 00:00:00.001

Device information

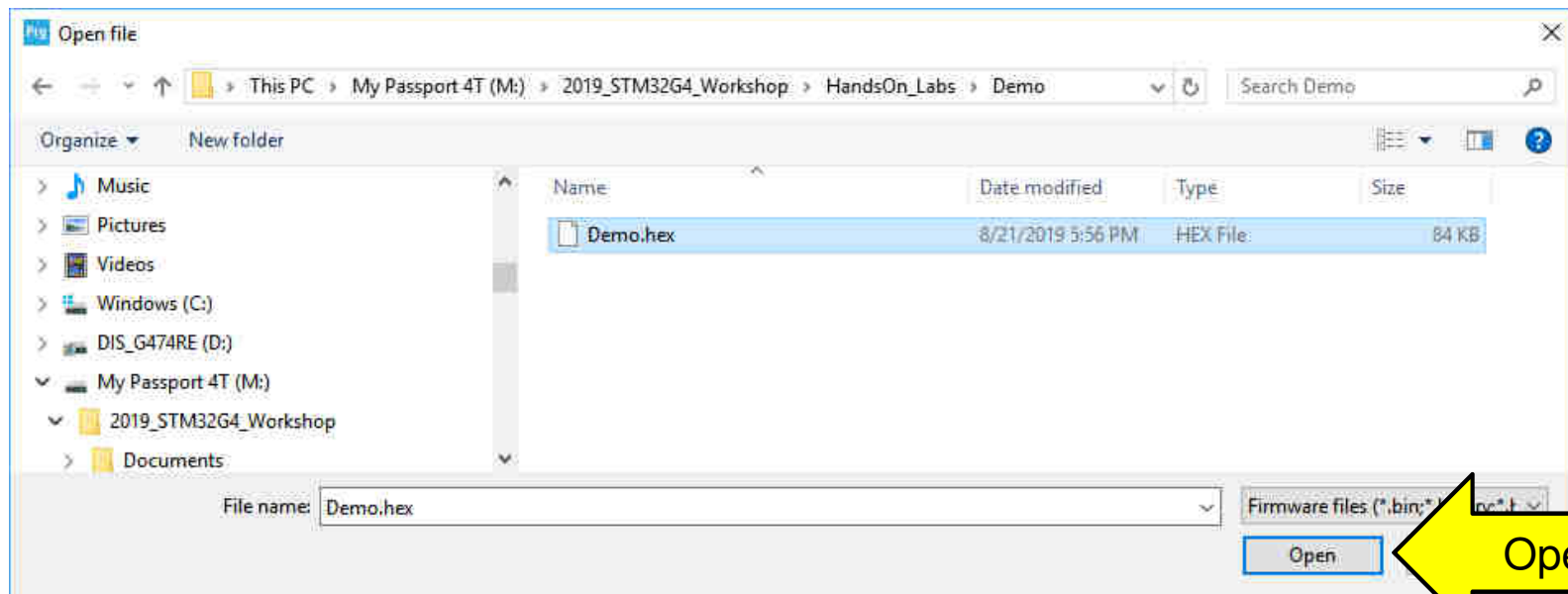
Device	STM32G47x/G48x
Type	MCU
Device ID	0x469
Flash size	512 KB
CPU	Cortex-M4



CubeProgrammer

37

File Folder





CubeProgrammer

38

Move Demo.hex from the PC to the Discovery board

The screenshot shows the STM32CubeProgrammer software interface. The 'Memory & File edition' window is open, displaying a table of memory addresses and their corresponding data. A yellow arrow points to the 'Download' button in the top right corner of the window.

Memory & File edition

Device memory: Demo.hex

Address: 0x8000000 Size: 0x400 Data width: 32-bit Download

Address	0	4	8	C	ASCII
0x08000000	20000968	080074D9	08007319	08007318	h..ùt...s...
0x08000010	0800731D	0800731F	08007321	00000000	.s...!s....
0x08000020	00000000	00000000	00000000	08007323	#s..
0x08000030	08007325	00000000	08007327	08007329	%s.....'s..)s..
0x08000040	080074F5	080074F9	080074FD	08007501	ôt..ùt..ýt...u..
0x08000050	08007505	08007509	0800750D	08007511	.u...u...u...u...
0x08000060	0800732D	08007515	08007333	08007519	-s...u...3s...u...
0x08000070	0800751D	08007521	08007525	08007529	.u...!u...%u...)u...
0x08000080	0800752D	08007531	08007535	08007539	-u...!u...5u...9u...
0x08000090	0800753D	08007541	08007545	08007339	=u...Au...Eu...9s...
0x080000A0	08007549	0800754D	08007551	08007555	Iu...Mu...Qu...Uu...
0x080000B0	08007559	0800755D	08007561	08007565	Yu...Ju...au...eu...
0x080000C0	08007569	0800756D	08007571	08007575	Iu...mu...qu...uu...

Log

12:58:46 : Address : 0x40022070
12:58:46 : Size : 8 Bytes
12:58:46 : UPLOADING...
12:58:46 : Size : 1024 Bytes
12:58:46 : Address : 0x8000000
12:58:46 : Read progress:
12:58:46 : Data read successfully
12:58:46 : Time elapsed during the read operation is: 00:00:00.001
13:51:18 : Read File: M:\2019_STM32G4_Workshop\HandsOn_Labs\Demo\Demo.hex
13:51:18 : Number of segments: 1
13:51:18 : ...segment[0]: address=0x8000000, size=0x76D9

Verbosity level: 1 2 3

Port: COM1
Frequency (KHz): 4000
Mode: Normal
Access point: 0
Reset mode: Software reset
Shared: Connected
External loader: ...
Target voltage: 3.29 V
Firmware version: V313M2
Firmware upgrade

Device information

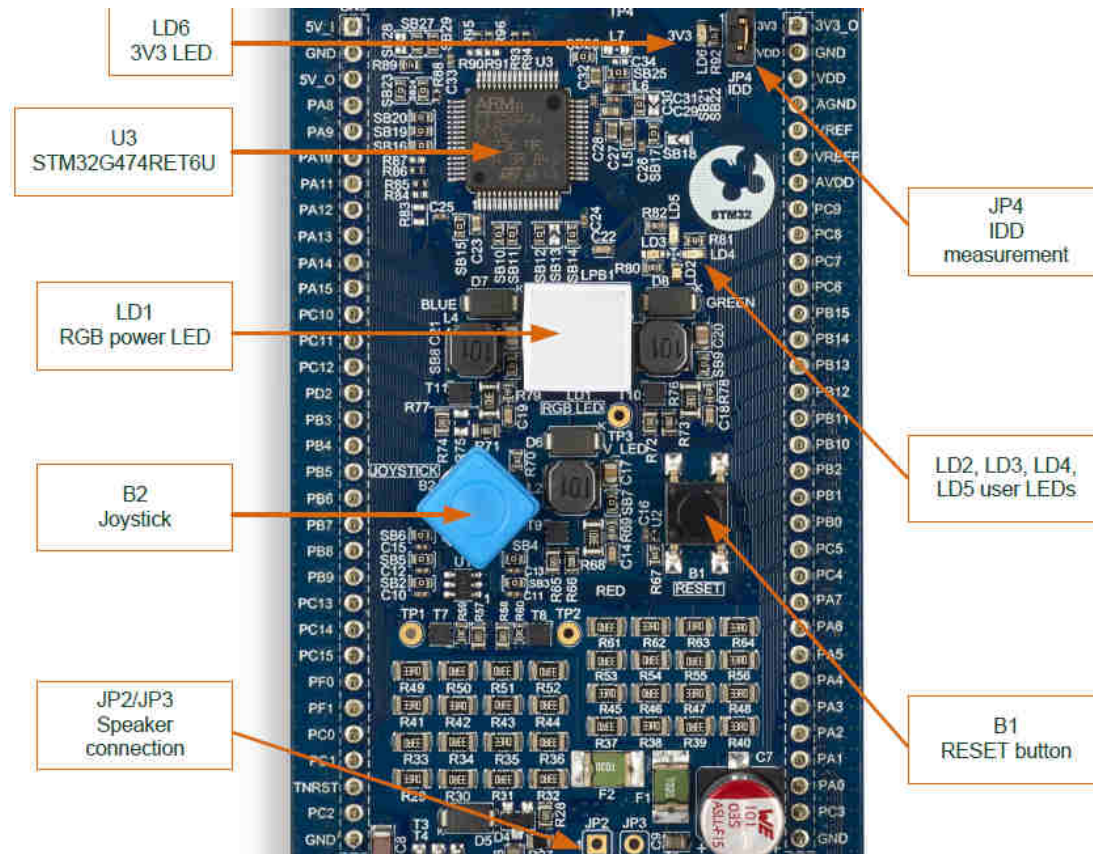
Device: STM32G47x/G48x
Type: MCU
Device ID: 0x469
Flash size: 512 KB
CPU: Cortex-M4



B-G474E-DPOW1

39

Press RESET (Black Button) – LED's cycle



LED's Cycle

PRESS Quickly

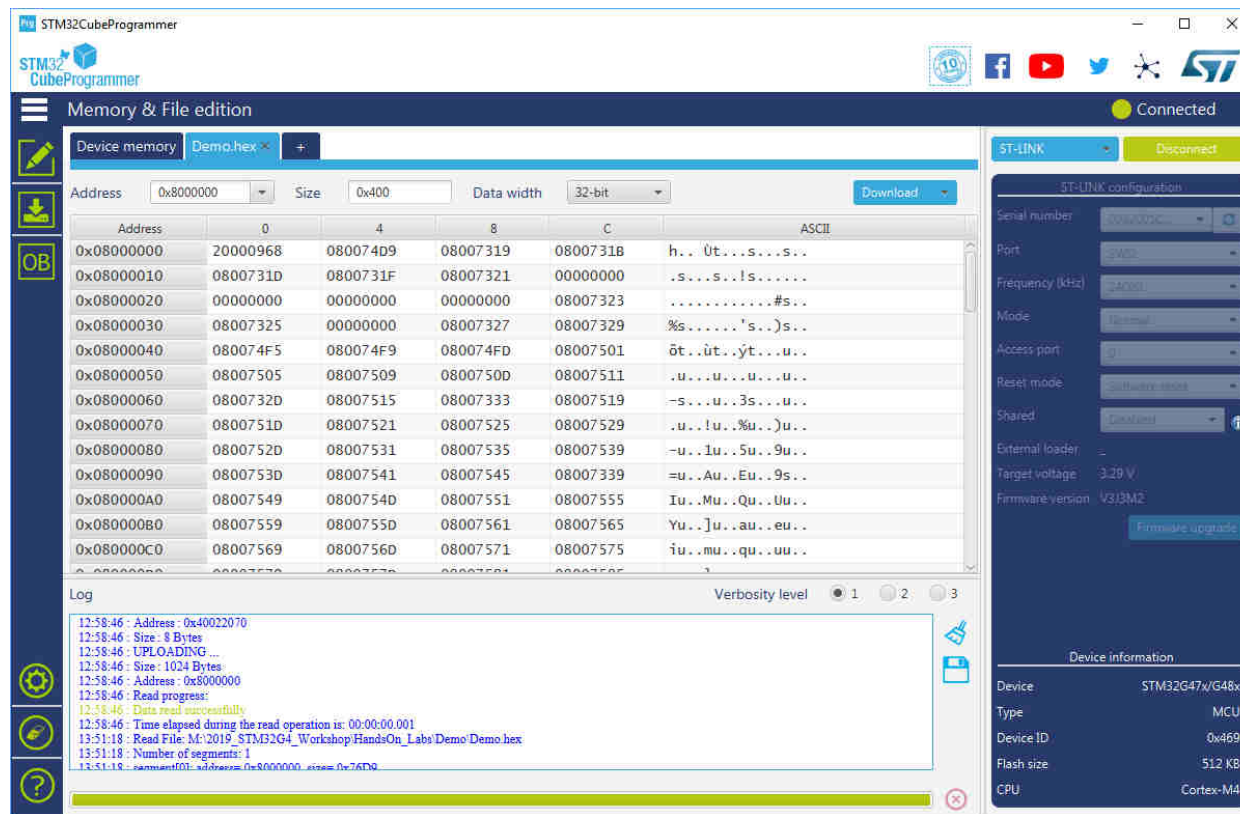


CubeProgrammer

40

Shut down the CubeProgrammer to release the VCOM port. ...

Exit





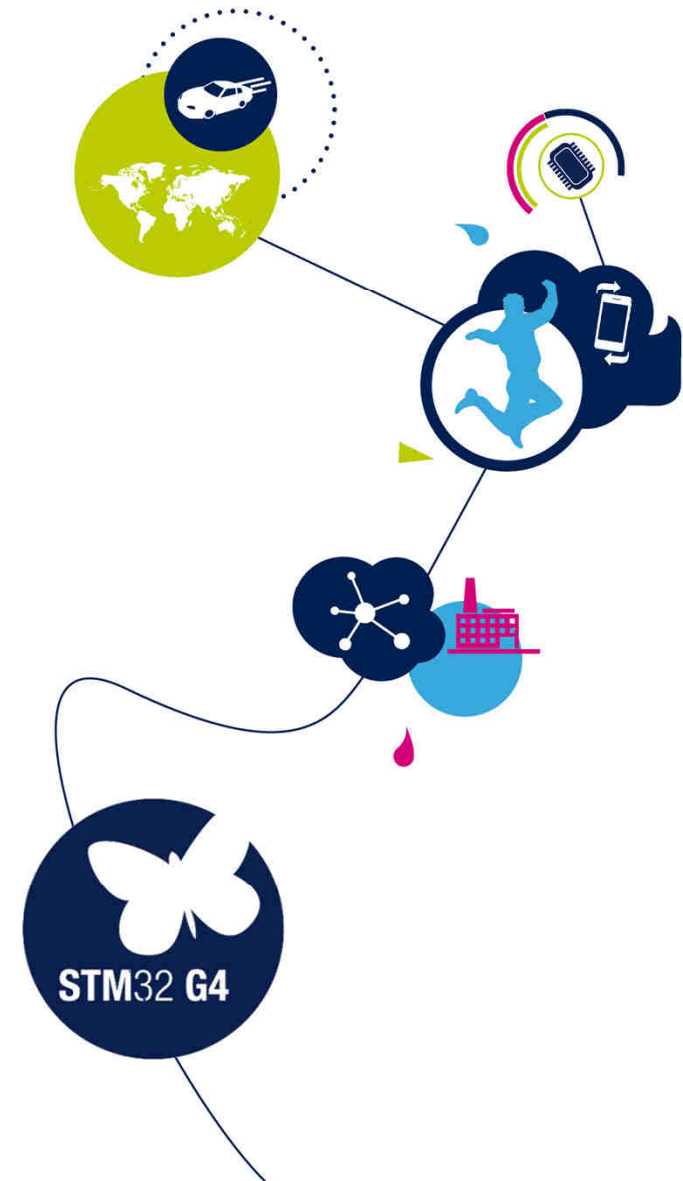
Getting back home – “CompletedLab”

For all of today’s Hands On Labs:

1. The *.hex file is available to you as a final product of the lab if you get lost along the way and still want to see the result of the lab.
2. The *.uvprojx file is available to you as a final result in case you want to pull it up in KEIL MDK-ARM and just walk through the files you see there.
3. The final STM32CubeMx *.ioc file for each lab is made available to you if you want to see the finished product of the lab or if you want to extend the lab with your own design ideas.

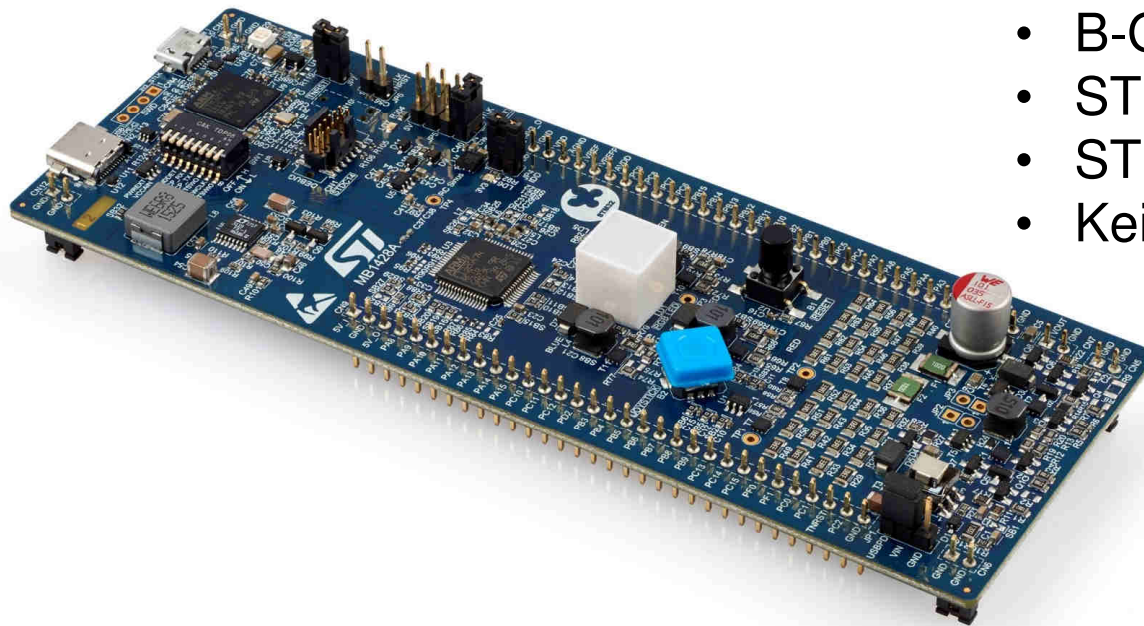
STM32G4 mixed-signal MCU workshop

*Hands On:
ADC Gain Compensation*





Systems Check



- USB cable
- B-G474E-DPOW1 Discovery Board
- STM32G4 Family Cube Library
- STM32CubeMx
- Keil MDK-ARM



Analog to Digital Converter

44

G4 ADC

- Up to 5 multi-input ADCs
- 4Msps - 12bit SAR
- 16bit hardware oversampling
- 2x to 256x oversample ratio
- DMA
- 3 Analog Watchdogs / ADC
- Single Ended or Differential inputs
- Independent clock

Internally connected to:

- Voltage reference
- Temperature Sensor
- 6 Op Amps
- Multiple Timers (+ HRTimer)



Analog to Digital Converter

45

New Features – Gain and Offset Compensation

Gain Compensation:

- Activated on all converted data
- Effective for oversampling
 - Calculation is done after accumulation
 - And after right-shifting

The lab will use this feature to show the voltage directly in millivolts without need for post computing of the converted data

Offset Compensation:

- Analog Watchdog performed on unsigned values after gain and offset compensation

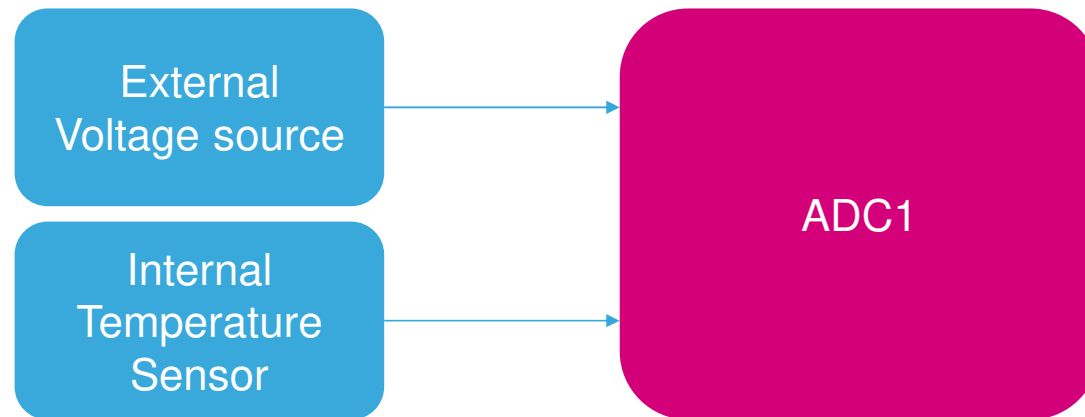
The lab directly translates the conversion data from the ADC range to the application range



Analog to Digital Converter

46

Block Diagram





ADC Gain Compensation

47

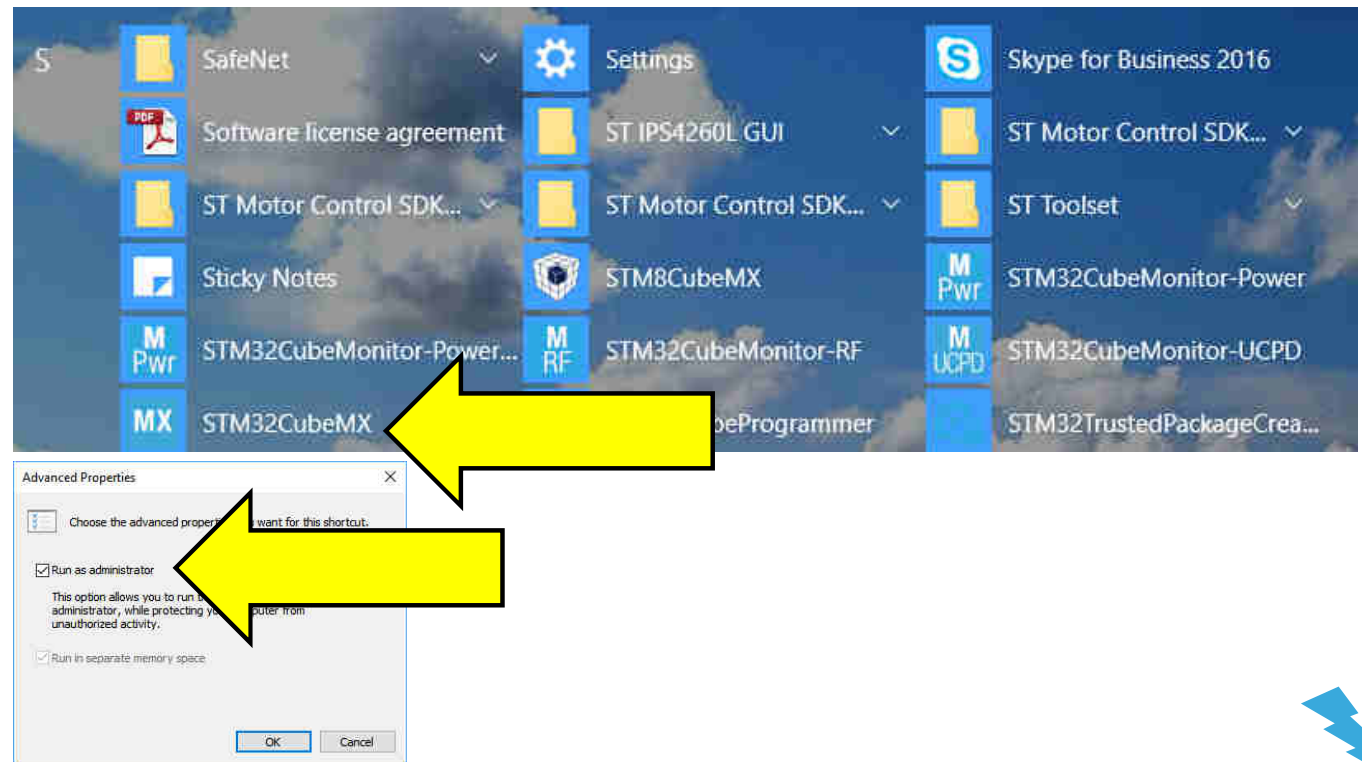
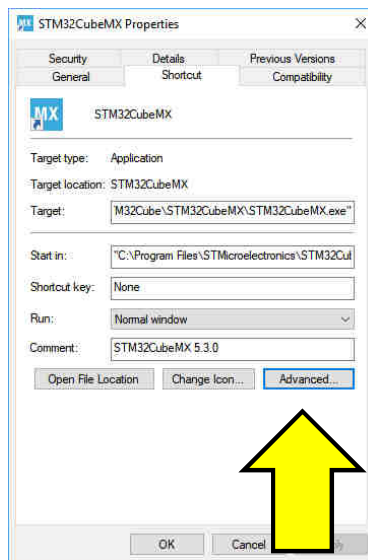
- Setting the GCOMP bit in ADC_CFGR2 activates the gain compensation
- The compensation is applied to all the converted data.
- After each conversion, data is calculated using the formula:

$$Data = Data(ADC_{Result}) \times \frac{(GCOMPCoefficient)}{4096}$$

Where the GCOMPCoefficient is a 14 bit value (16384).
Since this value is divided by 4096, the gain factor ranges from 0 to 3.999756



Start up





Splash Page – G4 Library Installed?

Embedded Software Packages Manager

STM32Cube MCU Packages and embedded software packs releases

Releases Information was last refreshed less than one hour ago.

Description	Installation	Available Version
STM32F7		
STM32G0		
STM32G4		
STM32Cube MCU Package for STM32G4 Series	1.1.0	1.1.0
STM32Cube MCU Package for STM32G4 Series	1.0.1	1.0.1

Details

From Local From Url Refresh Install Now Remove Now Close

Manage software installations

Check for STM32CubeMX and embedded software packa...

CHECK FOR UPDATES

Install or remove embedded software packa...

INSTALL / REMOVE

1. Install/Remove

2.

New multicore STM32MP1 Series for Industrial and IoT applications

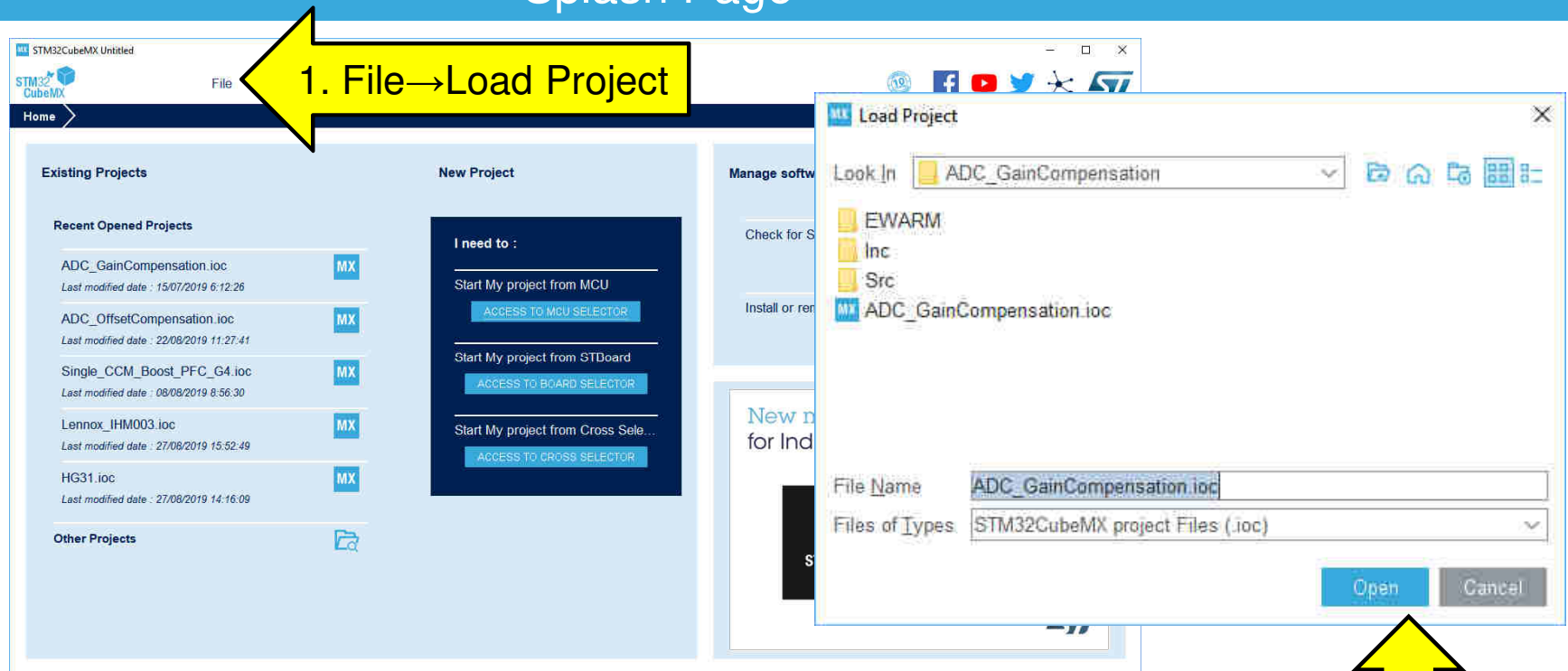
STM32MP1

OpenSTLinux Distribution

ST life.augmented



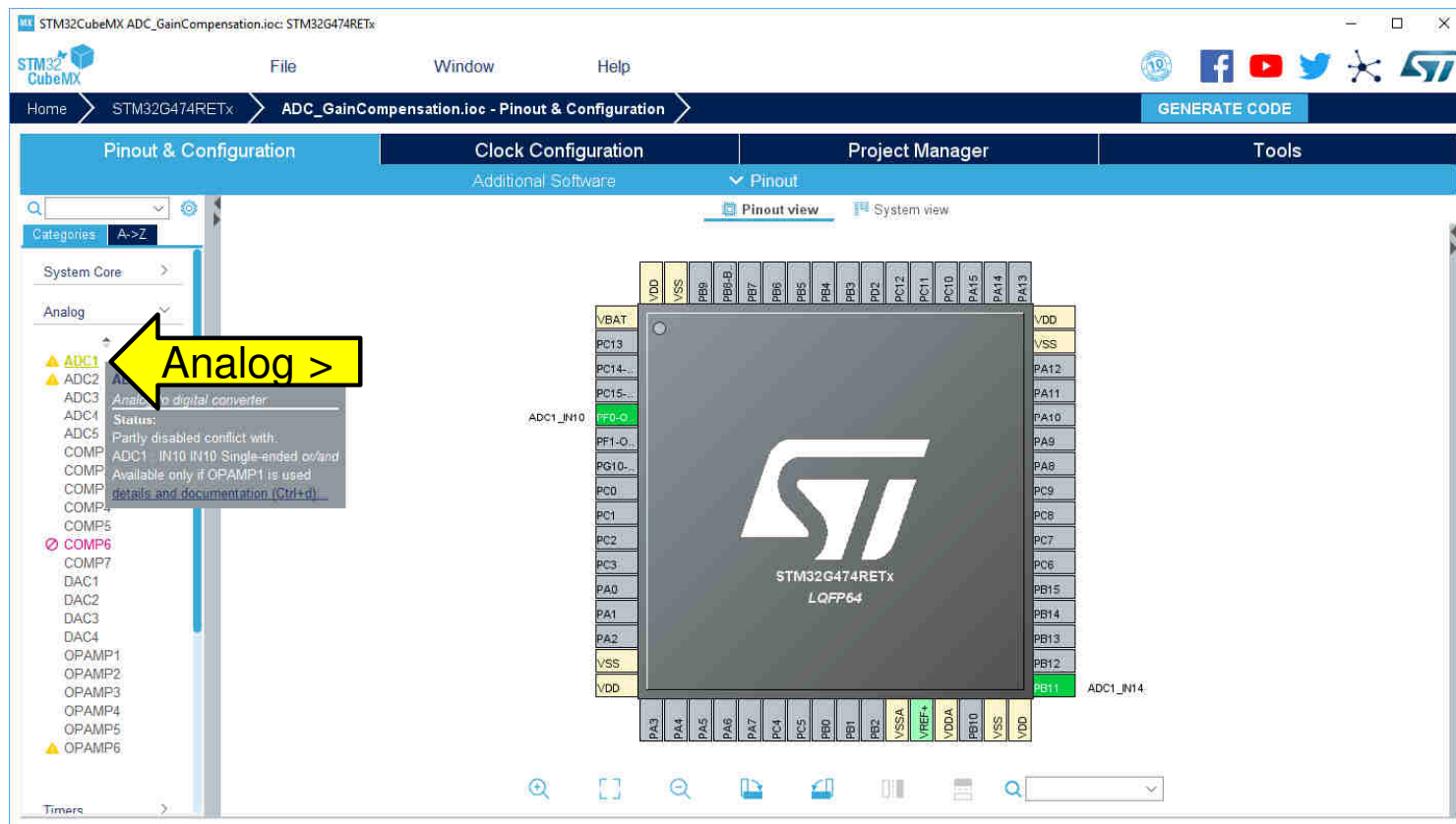
Splash Page



C:\...\STM32Cube\Repository\STM32Cube_FW_G4_V1.1.0\Projects
\B-G474E-DPOW1\Examples\ADC\ADC_GainCompensation



Graphic Design Representation





ADC1_IN10, ADC1_IN14, Gain Compensation

The screenshot displays the STM32CubeMX configuration tool for the STM32G474RETx microcontroller. The 'ADC1 Mode and Configuration' section is active, showing the following settings:

Parameter	Value
IN8	Disable
IN9	Disable
IN10	IN10 Single-ended
IN11	Disable
IN12	Single-ended
IN14	IN14 Single-ended
IN15	Single-ended

The 'Configuration' section shows the following parameters:

Parameter	Value
Clock Prescaler	Synchronous clock mode divided by 2
Resolution	ADC 12-bit resolution
Data Alignment	Right alignment
Gain Compensation	VDDA_APPLI * GAIN_COMPENSATION_X1_FACTOR / DIGITAL_SCALE_12BITS
Scan Conversion Mode	Disabled
End Of Conversion Selection	End of single conversion
Low Power Auto Wait	Disabled

The 'Gain Compensation' section provides a description: "Since GainCompensation is a parameter without check, you can enter any value. The default value is: 0."

The pinout diagram on the right shows the physical pins for ADC1_IN10 and ADC1_IN14, with a yellow arrow pointing to the 'Gain Compensation' parameter in the configuration table.



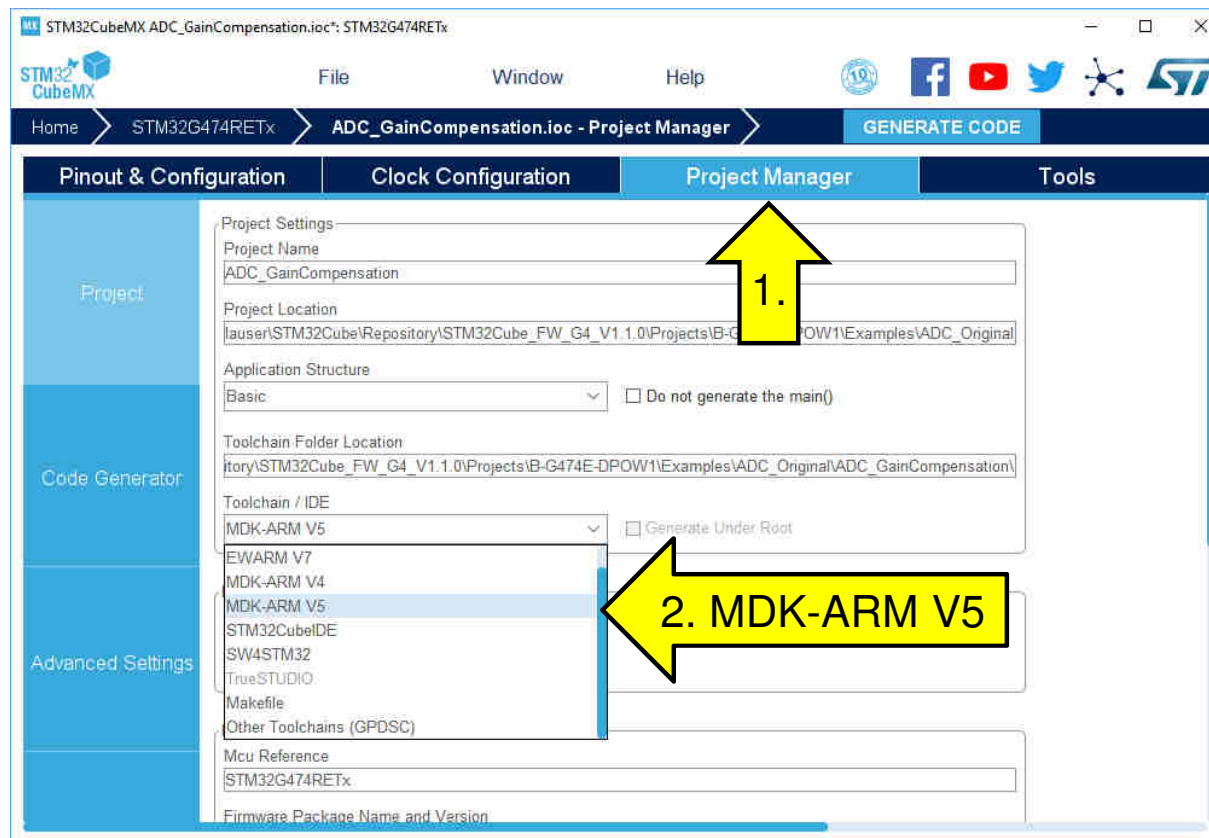
Gain Compensation Expanded

Data Alignment	Right alignment
Gain Compensation	VDDA_APPLI * GAIN_COMPENSATION_X1_FACTOR / DIGITAL_SCALE_12BITS
Scan Conversion Mode	Disabled

$$Data = Data(ADC_{Result}) \times \frac{(GCOMPCoeffcient)}{4096}$$

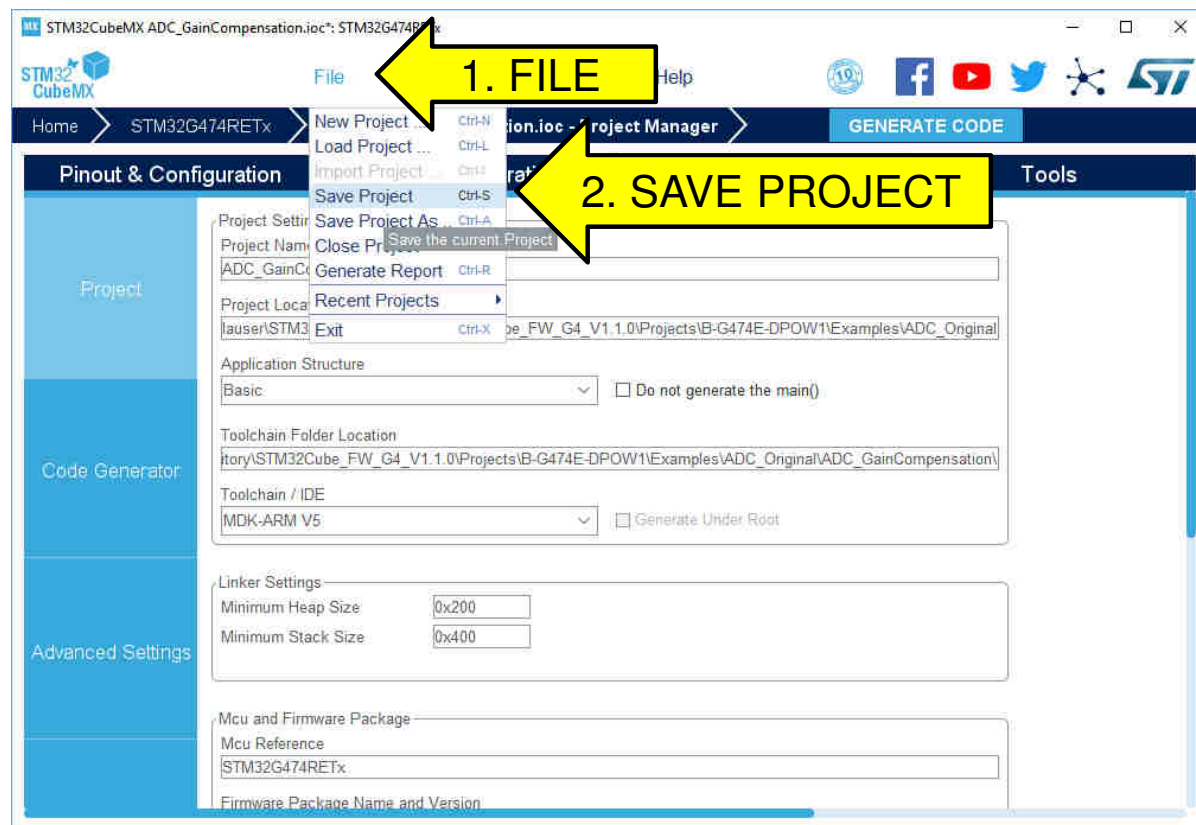


Project Manager – Keil MDK-ARM V5



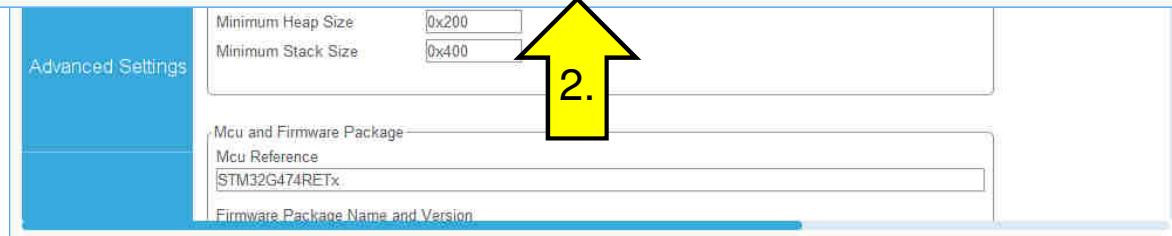
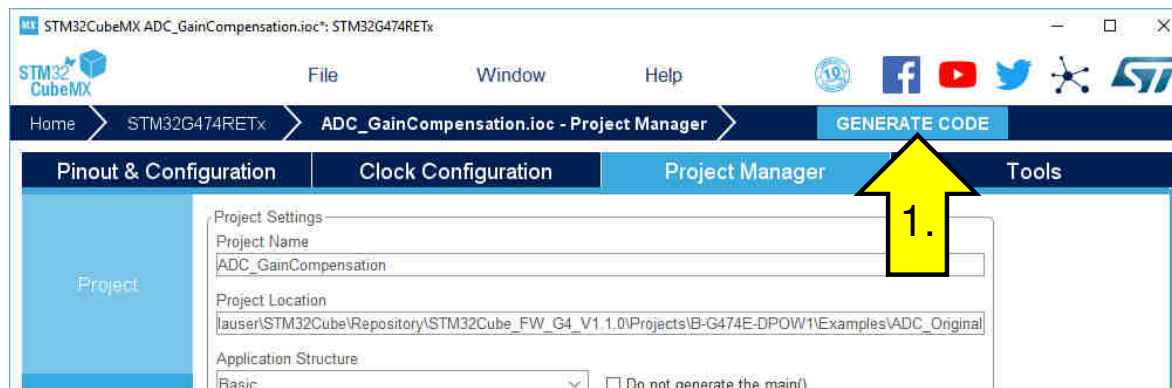


FILE→SAVE PROJECT



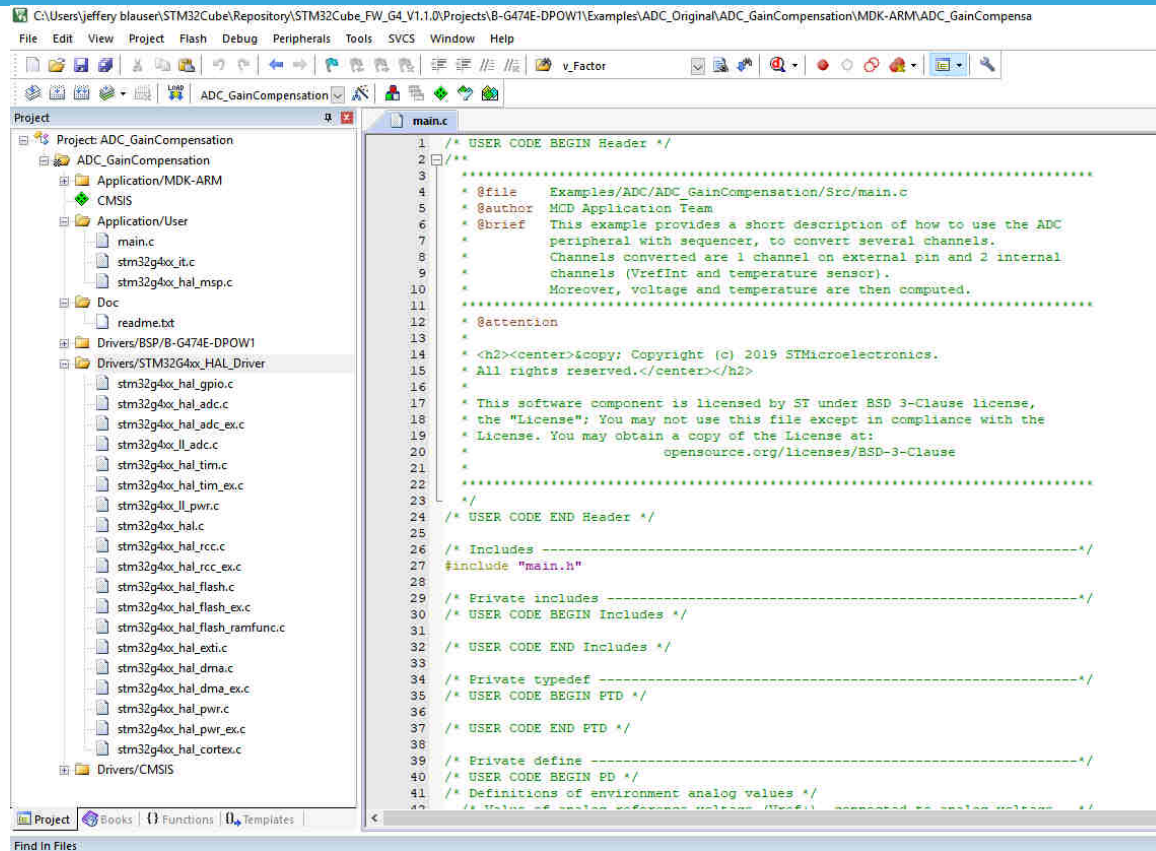


Generate Code





Keil Start up page

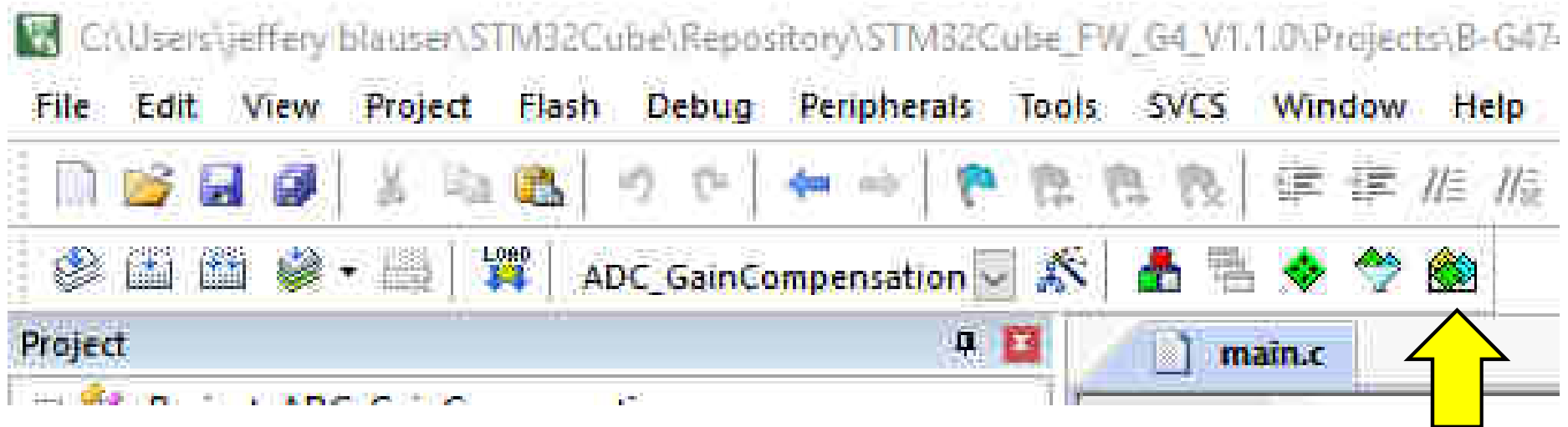




Keil MDK-ARM

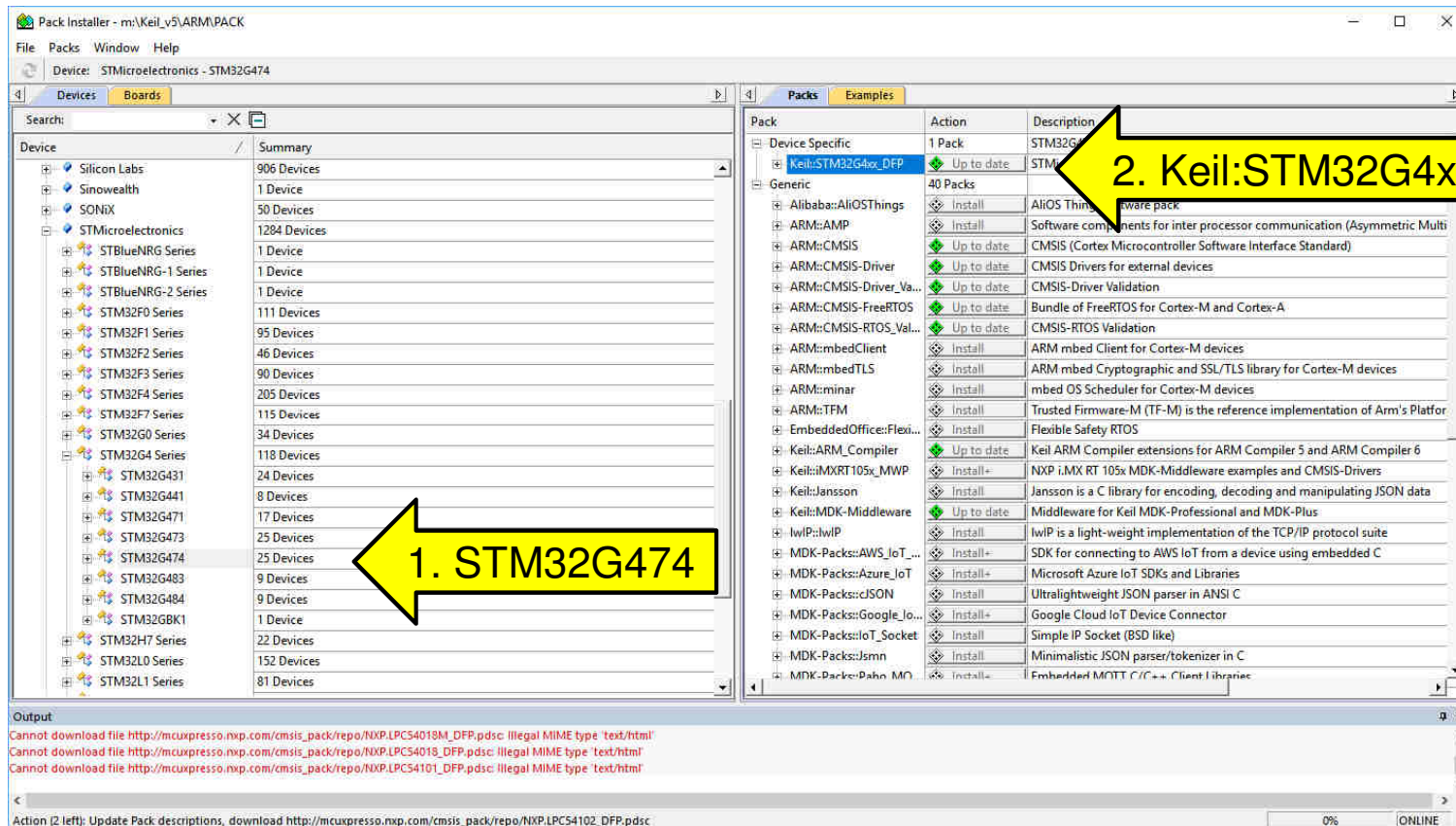
58

Check the Device Pack





G4 Device Pack

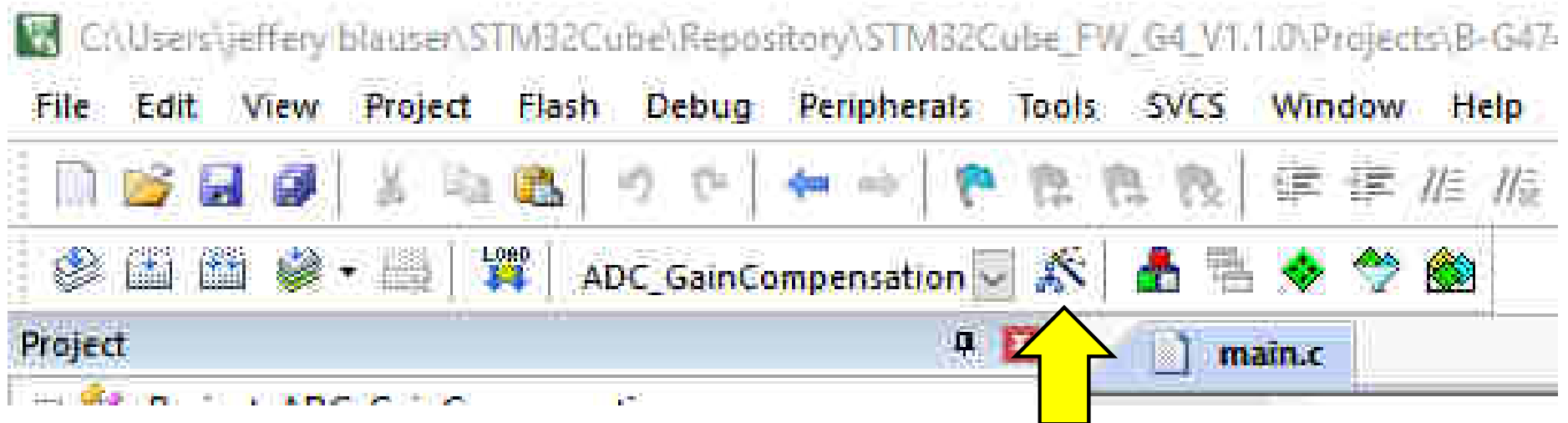




Keil MDK-ARM

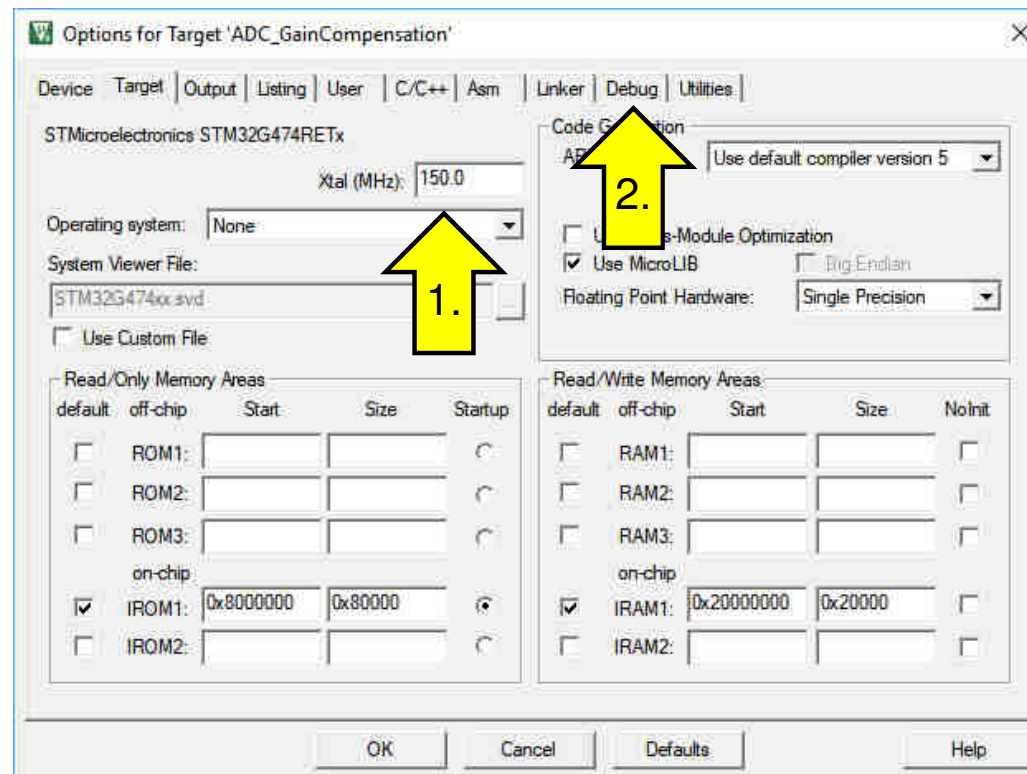
60

Check the Options



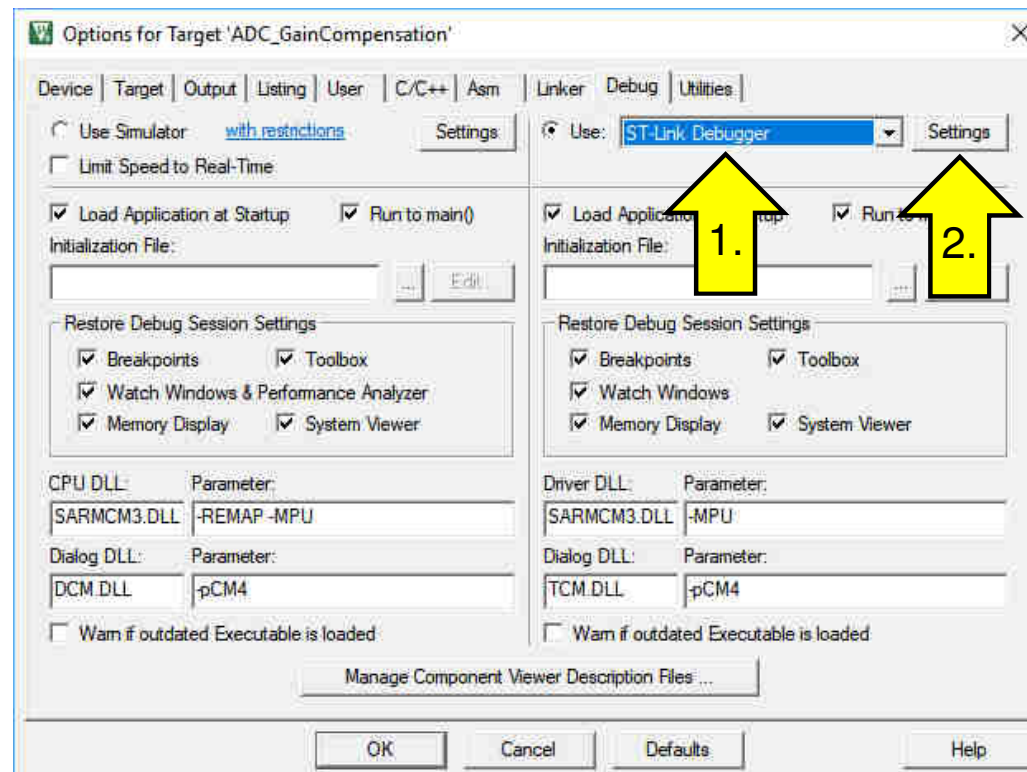


Clock speed and Debug



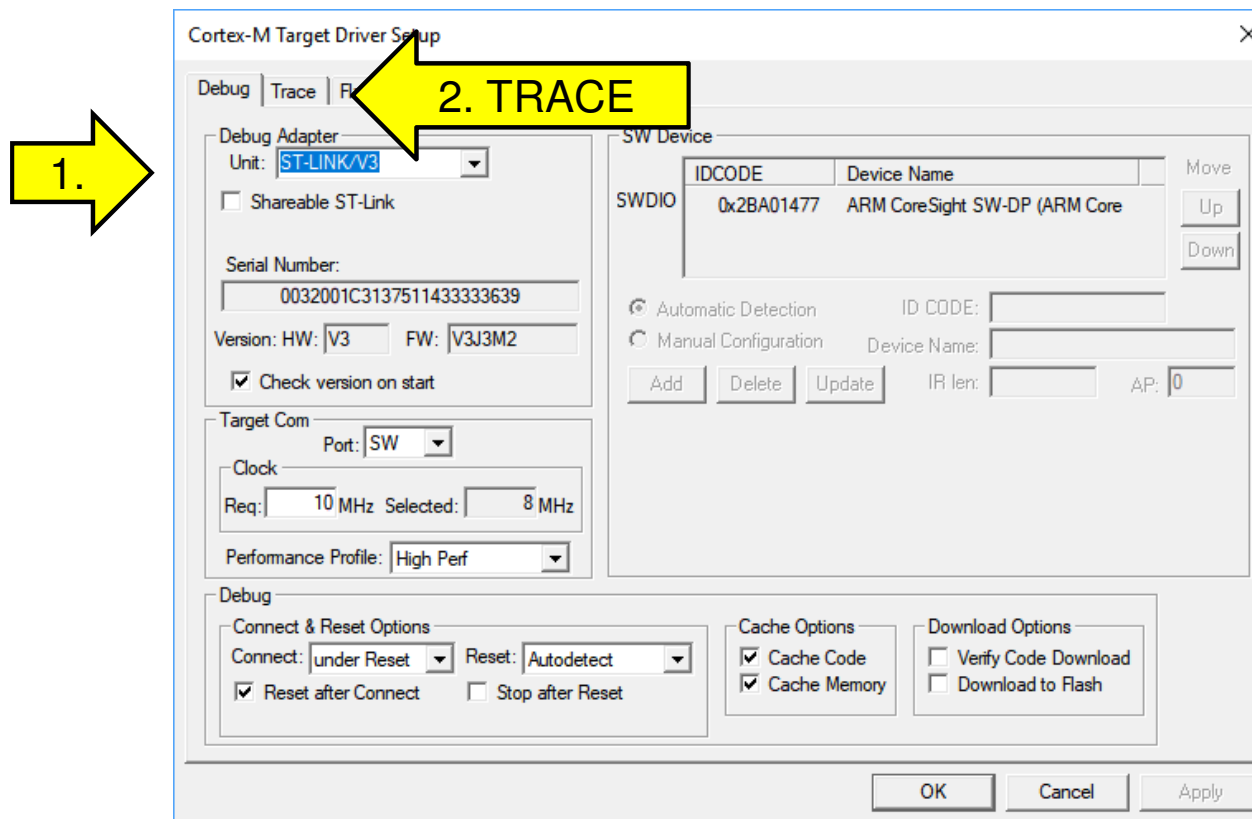


ST-Link Debugger - Settings





ST-Link/V3 & TRACE





Core Clock & TRACE Enable - Before

The screenshot shows the 'Cortex-M Target Driver Setup' dialog box with the 'Trace' tab selected. The following annotations are present:

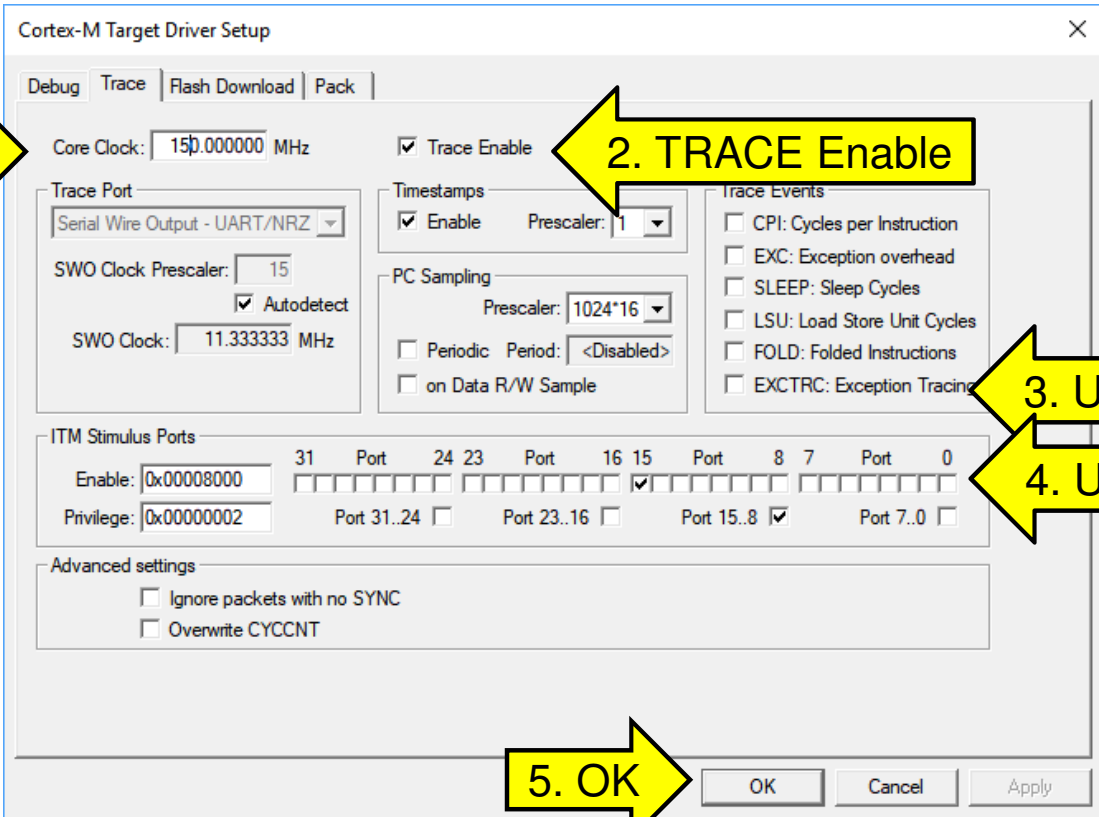
- 1. 150**: Points to the 'Core Clock' field, which is set to 150.000000 MHz.
- 2. TRACE Enable**: Points to the 'Trace Enable' checkbox, which is checked.
- 3. Uncheck EXCTRC**: Points to the 'EXCTRC: Exception Tracing' checkbox in the 'Trace Events' section, which is checked.
- 4. Uncheck all but 15**: Points to the 'ITM Stimulus Ports' section, specifically to the checkboxes for ports 31, 24, 23, 16, 15, 8, 7, and 0, which are all checked.

The dialog box contains the following sections:

- Debug** | **Trace** | **Flash Download** | **Pack**
- Core Clock**: 150.000000 MHz
- Trace Port**: Serial Wire Output - UART/NRZ
- SWO Clock Prescaler**: 13
- SWO Clock**: 11.538461 MHz
- Autodetect**: ☒
- Trace Enable**: ☒
- Timestamps**: ☒ Enable, Prescaler: 1
- PC Sampling**: Prescaler: 1024*16, ☐ Periodic, Period: <Disabled>, ☐ on Data R/W Sample
- Trace Events**:
 - ☐ CPI: Cycles per Instruction
 - ☐ EXC: Exception overhead
 - ☐ SLEEP: Sleep Cycles
 - ☐ LSU: Load Store Unit Cycles
 - ☐ FOLD: Folded Instructions
 - ☒ EXCTRC: Exception Tracing
- ITM Stimulus Ports**:
 - Enable: 0xFFFFFFFF
 - Privilege: 0x00000008
 - Port 31..24: ☒ Port 23..16: ☐ Port 15..8: ☐ Port 7..0: ☐
- Advanced settings**:
 - ☐ Ignore packets with no SYNC
 - ☐ Overwrite CYCCNT

Buttons: OK, Cancel, Apply

Core Clock & TRACE Enable - After



The screenshot shows the 'Cortex-M Target Driver Setup' dialog box with the 'Trace' tab selected. The following annotations highlight specific settings:

- 1. 150**: Points to the 'Core Clock' field, which is set to 150.000000 MHz.
- 2. TRACE Enable**: Points to the 'Trace Enable' checkbox, which is checked.
- 3. Uncheck EXCTRC**: Points to the 'EXCTRC: Exception Tracing' checkbox in the 'Trace Events' section, which is unchecked.
- 4. Uncheck all but 15**: Points to the 'Port 15..8' checkbox in the 'ITM Stimulus Ports' section, which is checked, while all other port checkboxes are unchecked.
- 5. OK**: Points to the 'OK' button at the bottom right of the dialog.

Additional visible settings include:

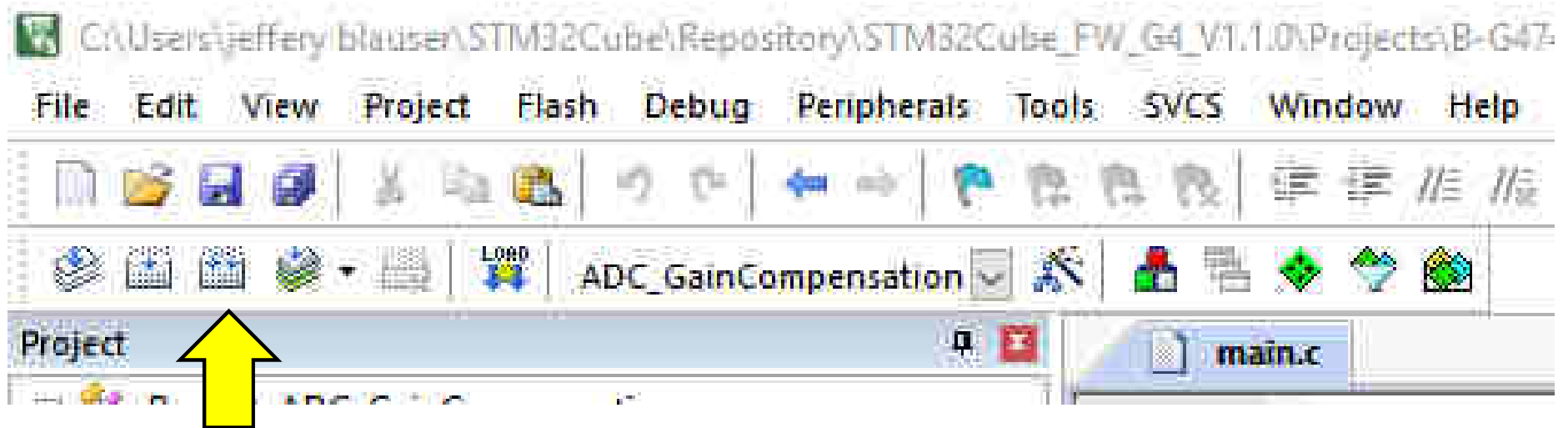
- Trace Port: Serial Wire Output - UART/NRZ
- SWO Clock Prescaler: 15
- SWO Clock: 11.333333 MHz
- Autodetect: checked
- Timestamps: Enable checked, Prescaler: 1
- PC Sampling: Prescaler: 1024*16, Periodic: disabled, on Data R/W Sample: unchecked
- ITM Stimulus Ports: Enable: 0x00008000, Privilege: 0x00000002
- Advanced settings: Ignore packets with no SYNC (unchecked), Overwrite CYCCNT (unchecked)



Keil MDK-ARM

66

Build





Error about line 233 in main.c

C:\Users\jeffery blausen\STM32Cube\Repository\STM32Cube_FW_G4_V1.1.0\Projects\B-G474E-DPOW1\Examples\ADC_Original\ADC_GainCompensation\MDK-ARM\ADC_GainCompensation

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

ADC_GainCompensation

Project: ADC_GainCompensation

- ADC_GainCompensation
 - Application/MDK-ARM
 - Application/User
 - main.c
 - stm32g4xx_it.c
 - stm32g4xx_hal_msp.c
 - Doc
 - readme.txt
 - Drivers/BSP/B-G474E-DPOW1
 - Drivers/STM32G4xx_HAL_Driver

```
226 /* USER CODE END ADC1_Init 1 */
227 /** Common config
228 */
229 hadcl.Instance = ADC1;
230 hadcl.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV2;
231 hadcl.Init.Resolution = ADC_RESOLUTION_12B;
232 hadcl.Init.DataAlign = ADC_DATAALIGN_RIGHT;
233 hadcl.Init.GainCompensation = VDDA_APPLI * V_FACTOR / DIGITAL_SCALE_12BITS;
234 hadcl.Init.ScanConvMode = ADC_SCAN_DISABLE;
235 hadcl.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
236 hadcl.Init.LowPowerAutoWait = DISABLE;
237 hadcl.Init.ContinuousConvMode = DISABLE;
238 hadcl.Init.NbrOfConversion = 1;
```

Build Output

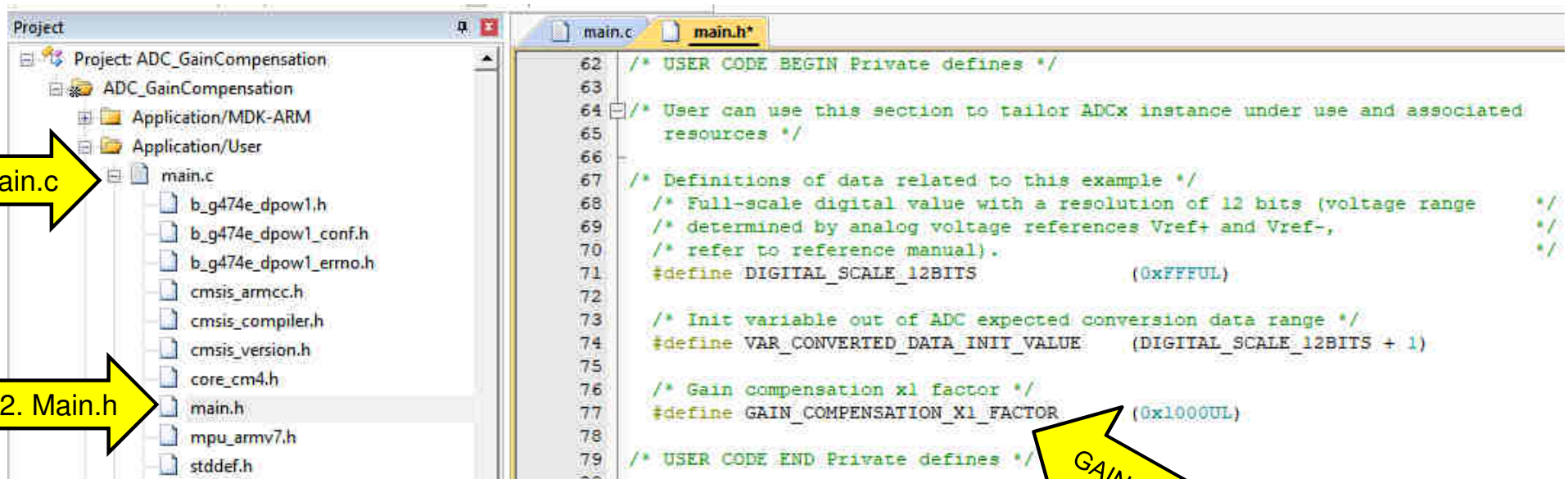
```
Rebuild target 'ADC_GainCompensation'
assembling startup_stm32g474xx.s...
compiling main.c...
../Src/main.c(233): error: #20: identifier "V_FACTOR" is undefined
    hadcl.Init.GainCompensation = VDDA_APPLI * V_FACTOR / DIGITAL_SCALE_12BITS;
../Src/main.c: 0 warnings, 1 error
```

1. V_FACTOR

2.



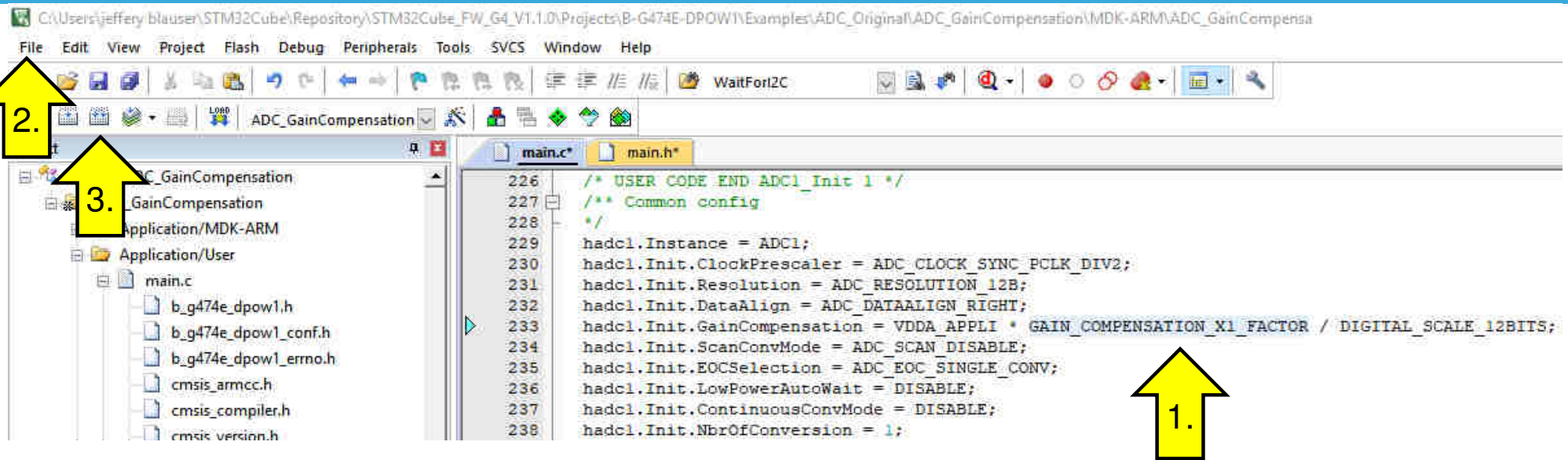
About line 77 in main.h – Copy - GAIN_COMPENSATION_X1_FACTOR



GAIN_COMPENSATION_X1_FACTOR



Paste into line 233 in main.c, FILE→SAVE, BUILD

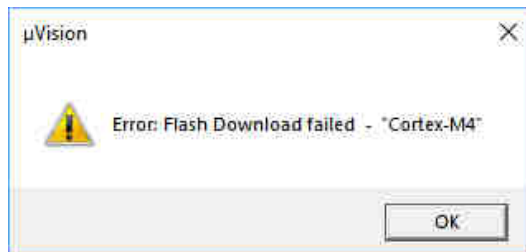
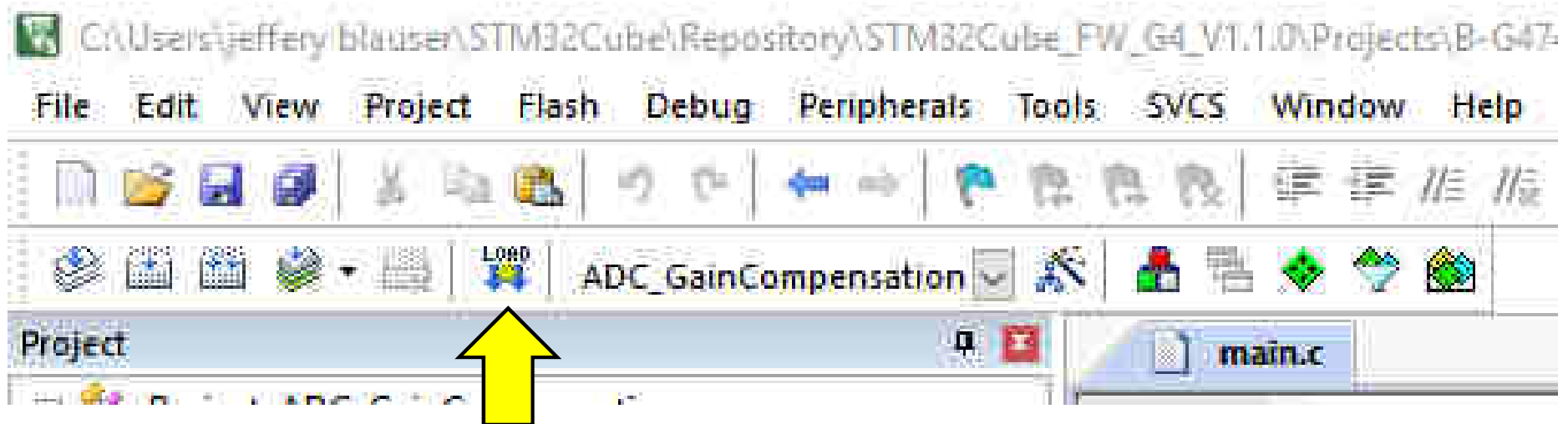


Build Output

```
compiling system_stm32g4xx.c...
linking...
Program Size: Code=9444 RO-data=576 RW-data=80 ZI-data=1296
FromELF: creating hex file...
ADC_GainCompensation\ADC_GainCompensation.hex: Warning: Q9931W: Your license for feature Keil will expire in 30 days
Finished: 0 information, 1 warning and 0 error messages.
"ADC_GainCompensation\ADC_GainCompensation.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:01:18
```



Load



Build Output

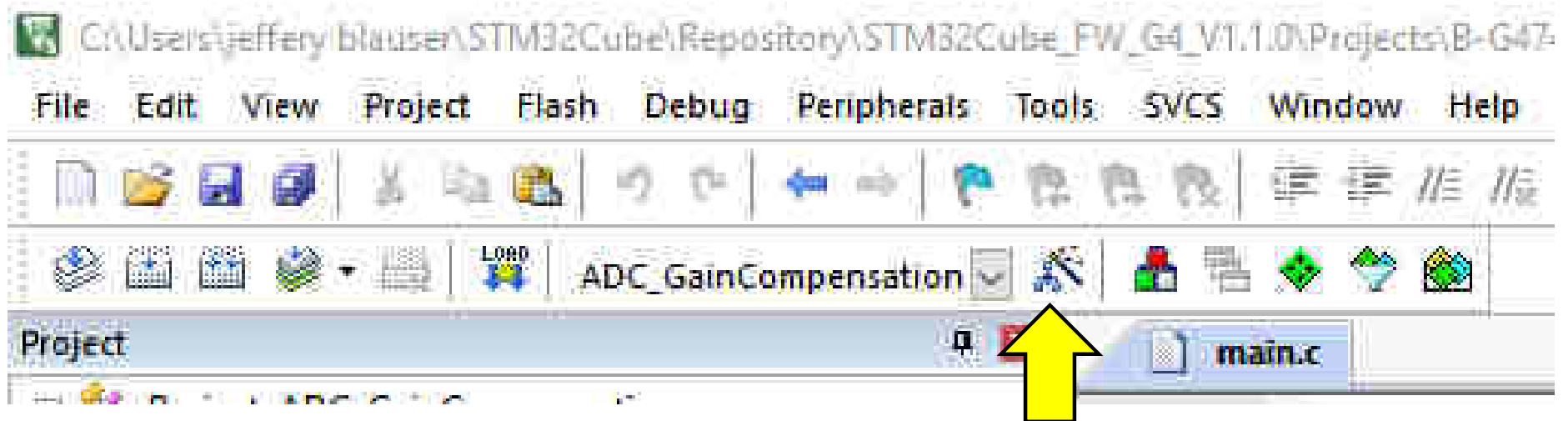
```
Contents mismatch at: 08001860H (Flash=38H Required=C5H) !
Contents mismatch at: 08001861H (Flash=46H Required=F8H) !
Contents mismatch at: 08001862H (Flash=FEH Required=88H) !
Contents mismatch at: 08001863H (Flash=F7H Required=00H) !
Contents mismatch at: 08001864H (Flash=B9H Required=60H) !
Contents mismatch at: 08001865H (Flash=FCH Required=6CH) !
Too many errors to display !
Error: Flash Download failed - "Cortex-M4"
```



Keil MDK-ARM

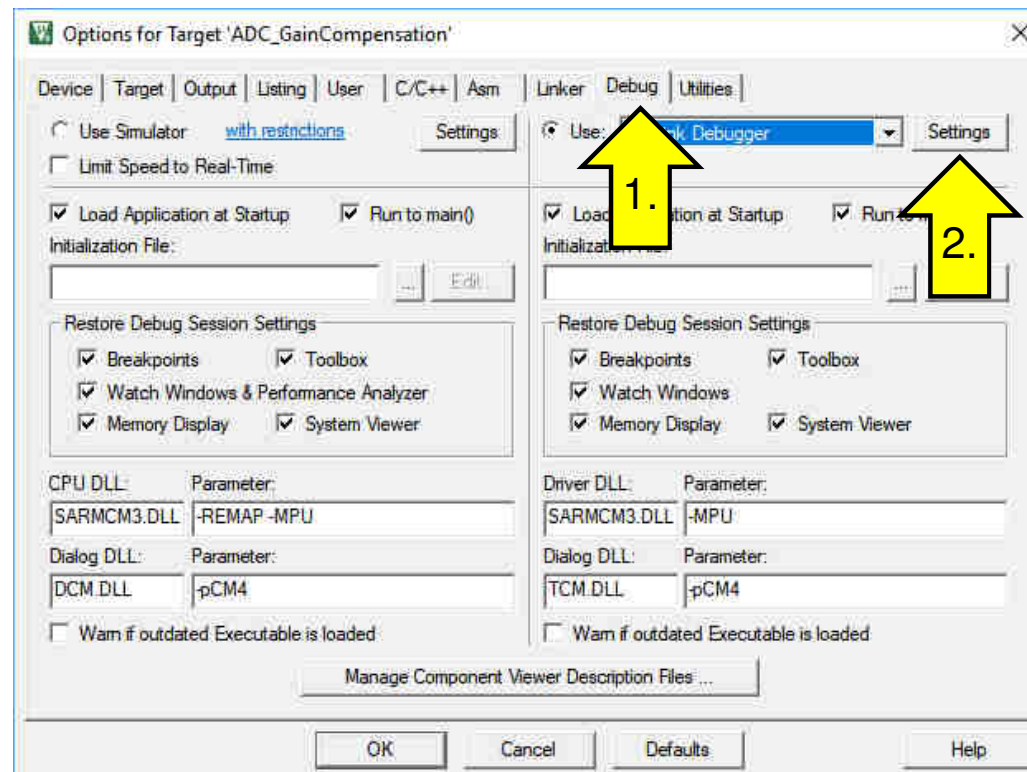
71

Options



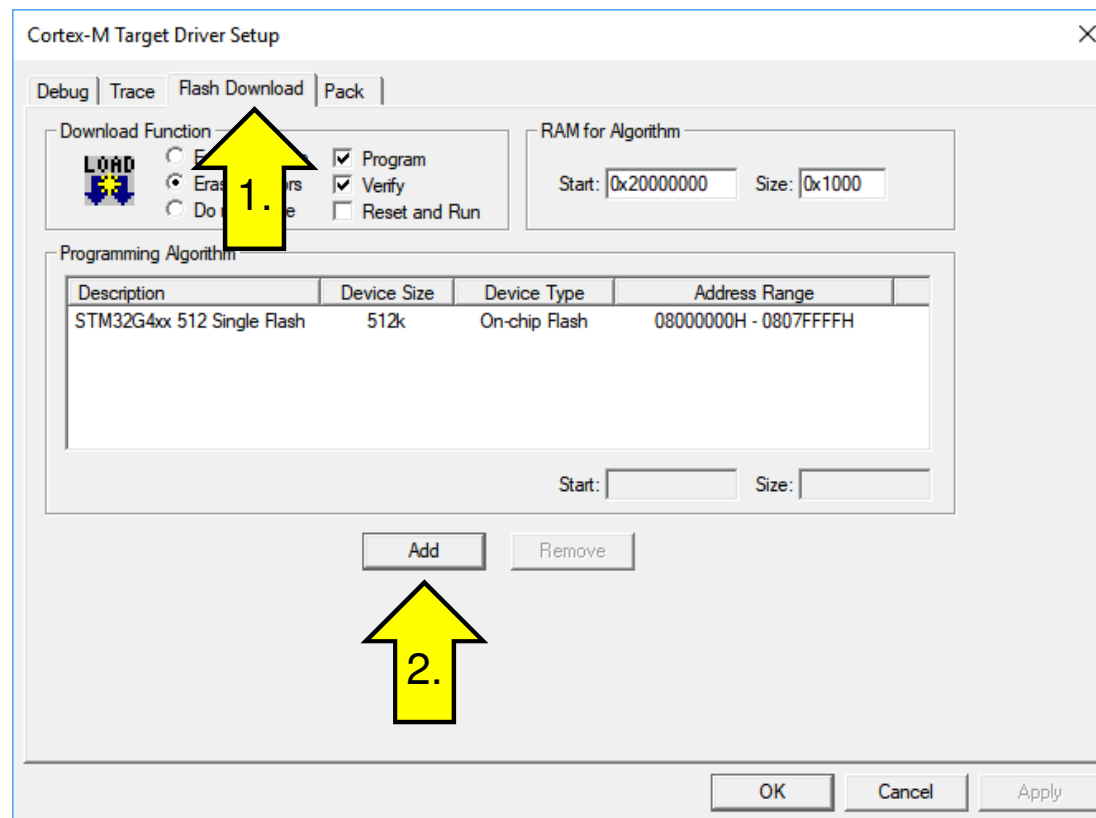


ST-Link Debugger - Settings



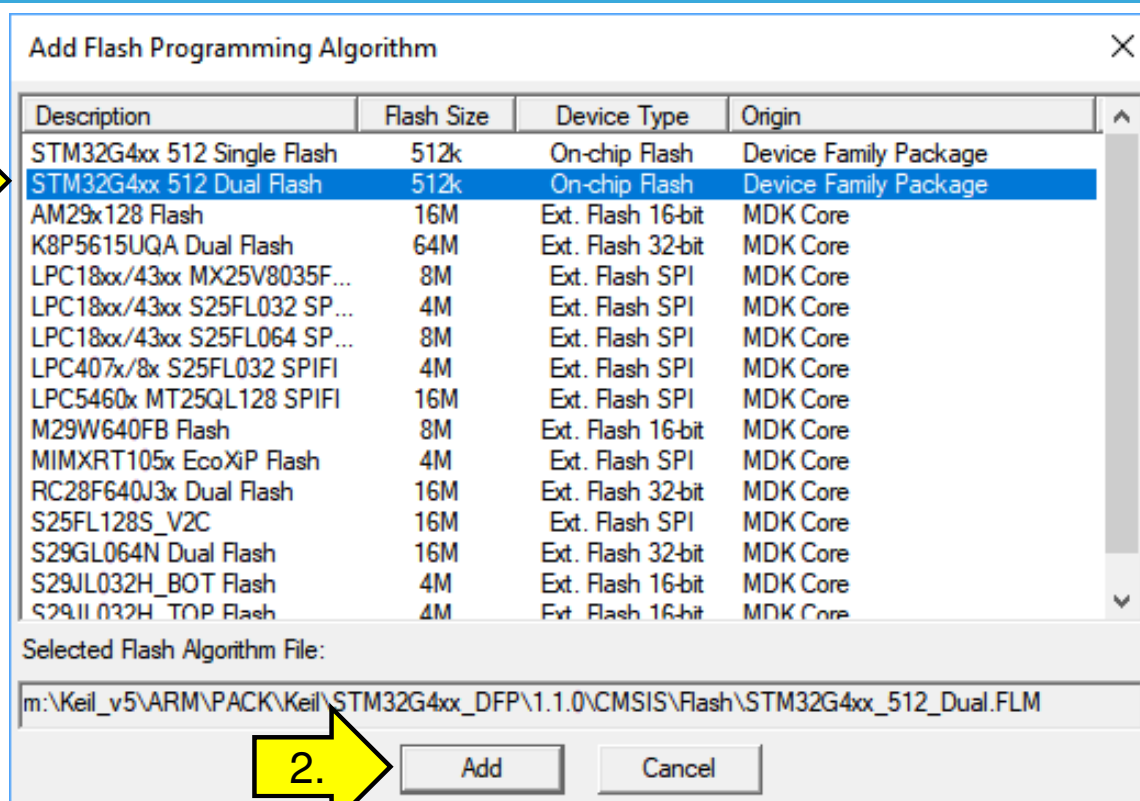


Flash Download → Add

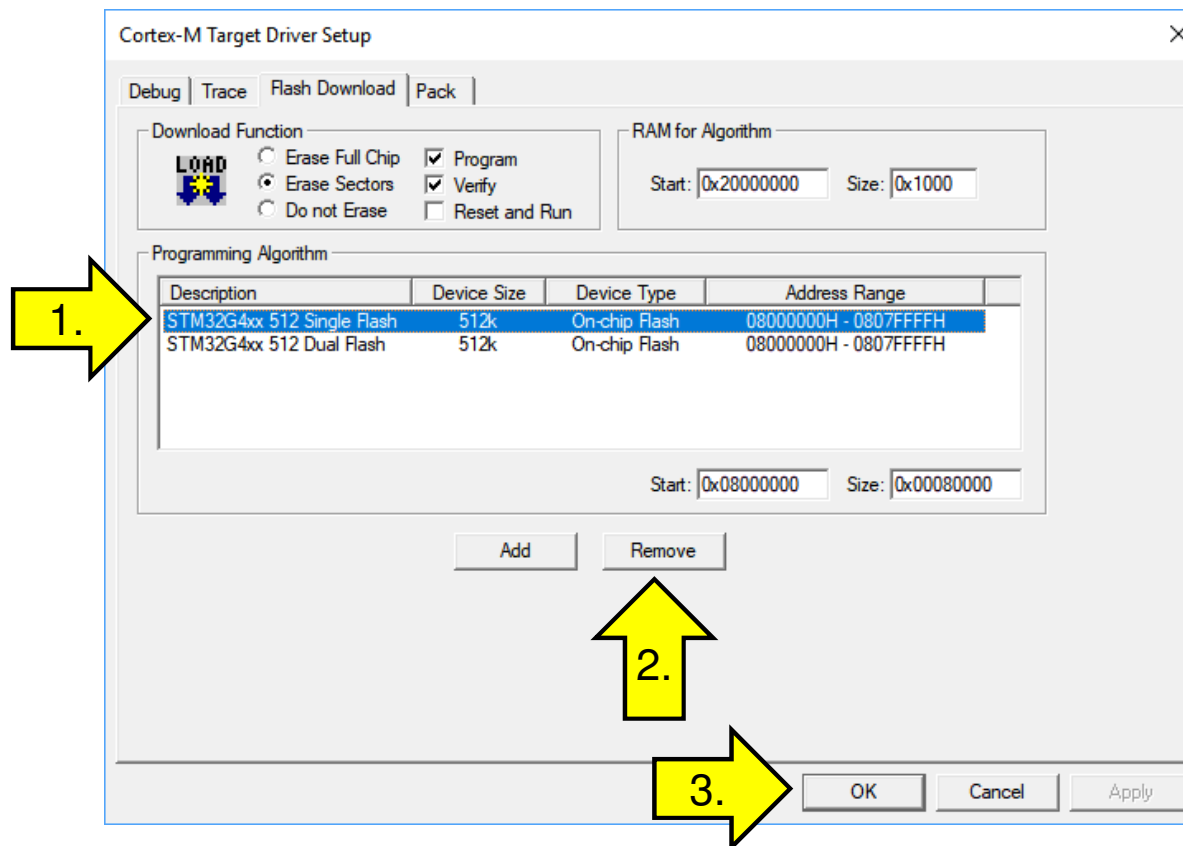




STM32G4xx 512 Dual Flash

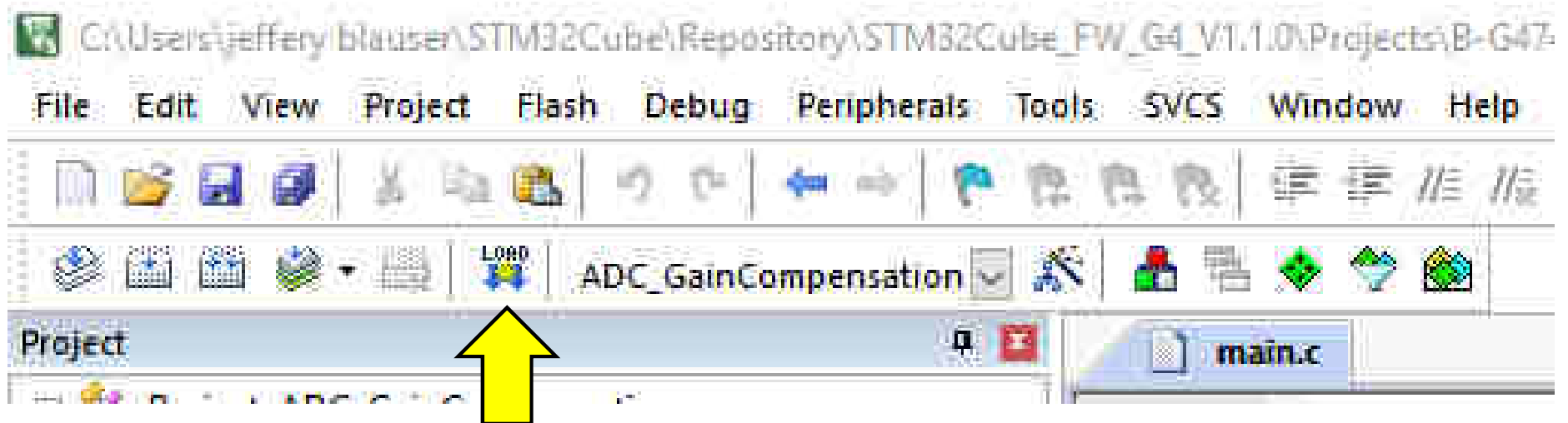


Remove Single Flash





Load



Build Output

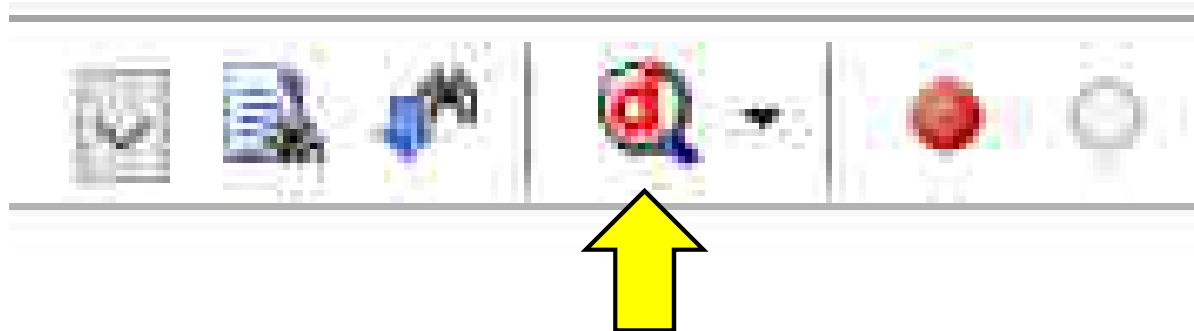
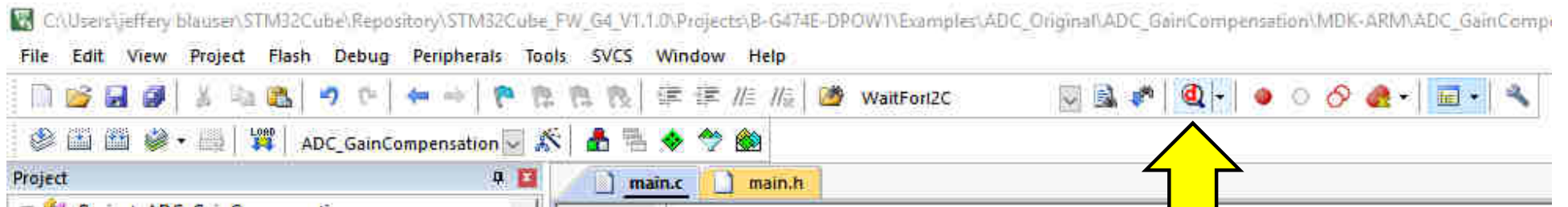
```
Flash Load finished at 07:02:32
Load "ADC_GainCompensation\\ADC_GainCompensation.axf"
Erase Done.
Programming Done.
Verify OK.
Flash Load finished at 07:12:10
<
```



Keil MDK-ARM

77

Debugger





Add 2 Lines - main.c about line 153

```
148 while (1)
149 {
150     /* USER CODE END WHILE */
151
152     /* USER CODE BEGIN 3 */
153     Display_ADC_Data_mVolt=uhADCxConvertedData_Voltage_mVolt; //jsb just for display
154     HAL_Delay(1); //jsb 1mS update rate
155 }
156 /* USER CODE END 3 */
```

Line 153 is a global variable for use by the Logic Analyzer

Line 154 is set to 1 millisecond to pull this data from the micro to the PC for display in the Logic Analyzer.



See c:\2019_STM32G4_Workshop\HandsOn_Labs\ADC_GainCompensation
Then *CodeLinesForMain_ADC_GainCompensation.txt*



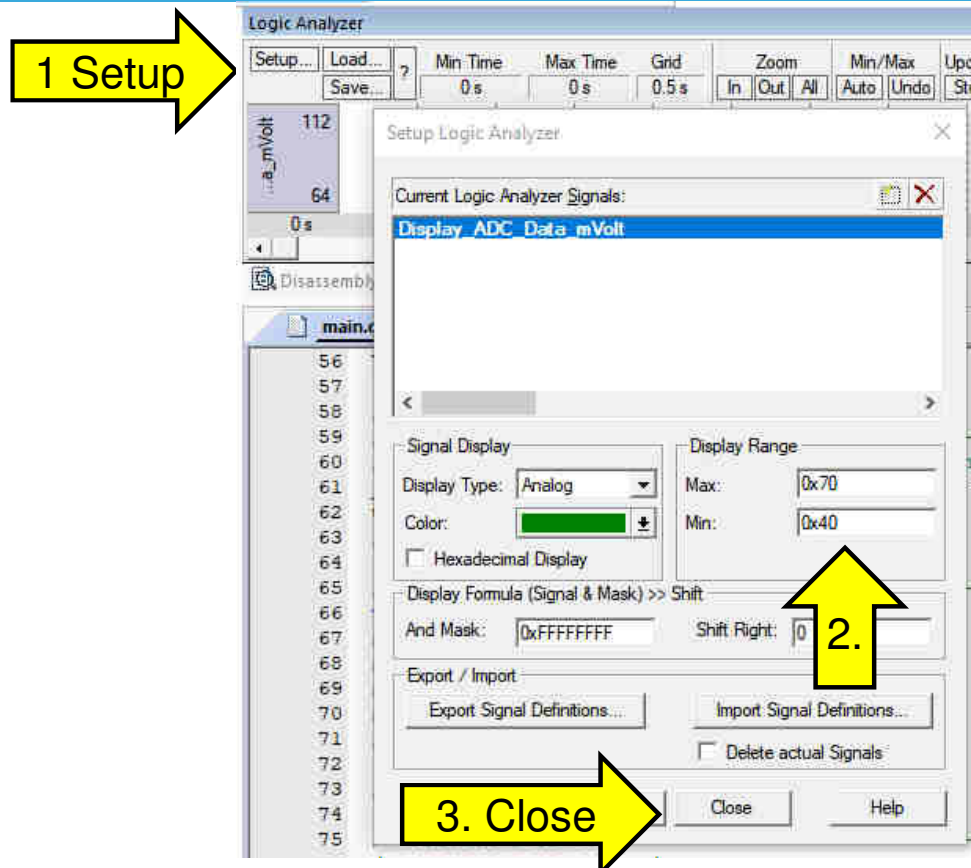
Add Line - Place variable in the Analyzer – main.c about line 62

```
58 /* USER CODE BEGIN PV */
59 /* Private variables ----- */
60 /* Variables for ADC conversion data computation to physical values */
61 IO uint16_t uhADCxConvertedData_Voltage_mVolt = 0U; /* Value of voltage on GPIO pin (on w
62 uint16_t Display_ADC_Data_mVolt; //
63 /* USER CODE END PV */
64
65 /* Private function prototypes
66 void SystemClock_Config(void);
67 static void MX_GPIO_Init(void);
68 static void MX_DMA_Init(void);
69 static void MX_ADC1_Init(void);
70 static void MX_TIM1_Init(void);
71 /* USER CODE BEGIN PFP */
72
73 /* USER CODE END PFP
74
75 /* Private user code -----
76 /* USER CODE BEGIN 0 */
77
78 /* USER CODE END 0 */
79
80 /**
81  * @brief The application en
82  * @retval int
83  */
```

1.

2.

Setup – Min=0x40, Max = 0x70

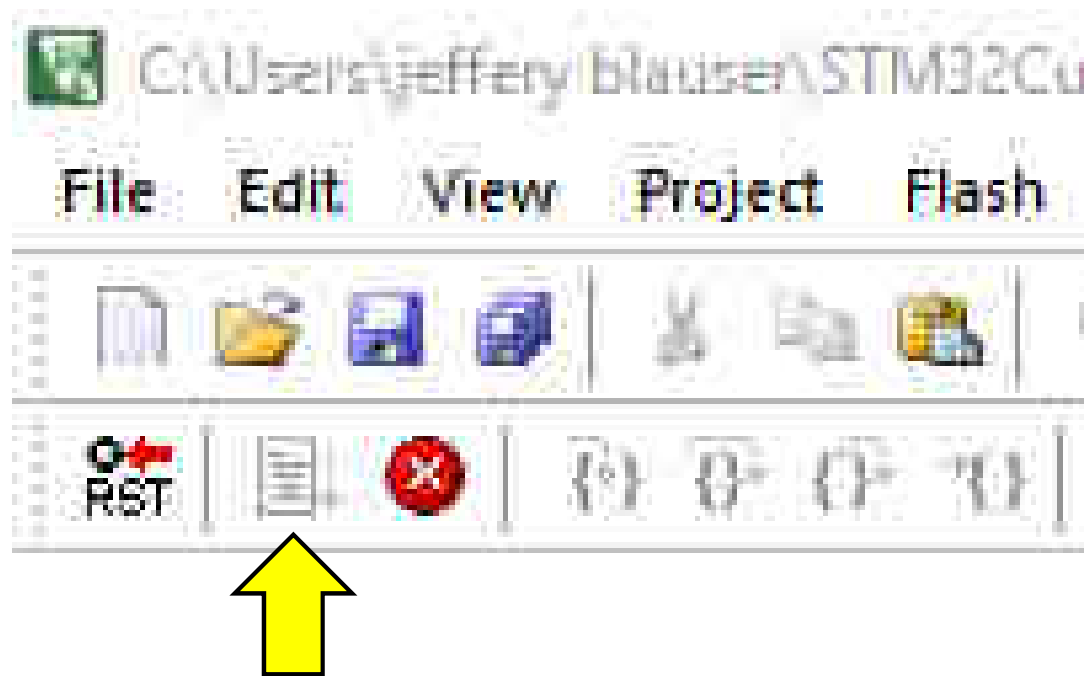




Keil MDK-ARM

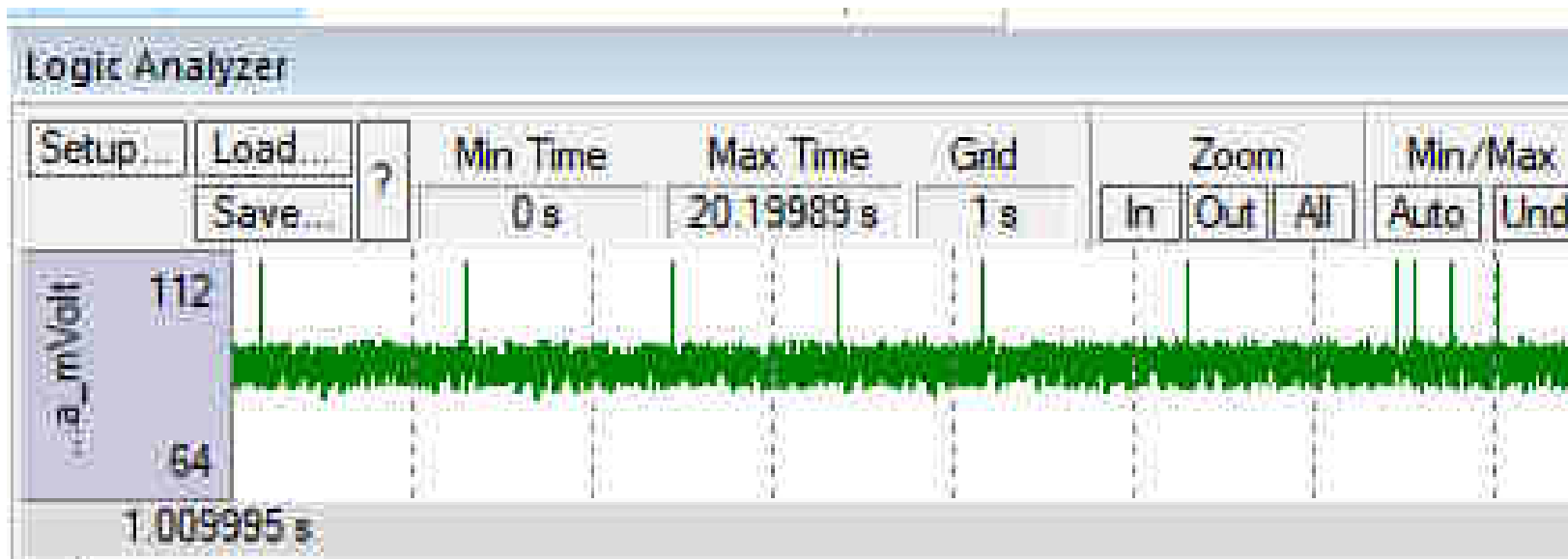
81

RUN



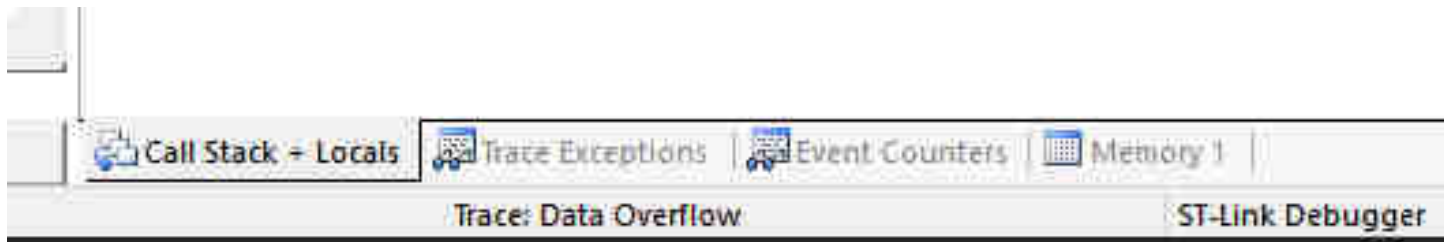


Logic Analyzer Output – Noise at the ADC input





Trace Overflow



1. ADC_Trigger source = _____ {which peripheral}
2. ADC_Trigger source clock speed = _____ MHz
3. ADC_Trigger source prescaler = _____
4. ADC_Trigger Speed = _____ Hz
5. ADC Resolution = _____ bits
6. SWO data throughput for ADC data = _____



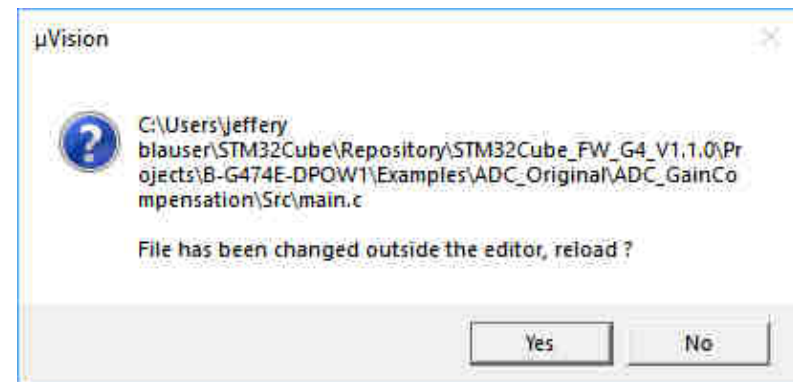
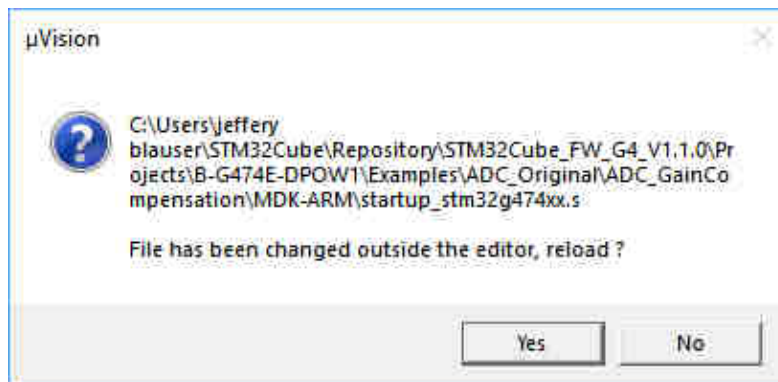
New Signal Source – Please do these steps in order

The screenshot shows the STM32CubeMx interface for configuring ADC1. The left sidebar lists components under 'Analog' and 'Timers'. The main window is titled 'ADC1 Mode and Configuration' and is divided into 'Mode' and 'Configuration' sections. Yellow arrows and text boxes indicate the following steps:

- 1. ADC1**: Points to the 'ADC1' entry in the 'Analog' category of the left sidebar.
- 2. Temp Sensor**: Points to the 'Temperature Sensor Channel' checkbox in the 'Mode' section.
- 3. Temp Sensor**: Points to the 'Channel Temperature Sensor' option in the 'Configuration' section, under 'ADC_Injected_ConversionMode'.
- 4. Disable IN10**: Points to the 'IN10 Single-ended' dropdown menu in the 'Mode' section.
- 5. Disable IN14**: Points to the 'IN14 Single-ended' dropdown menu in the 'Mode' section.
- 6. FILE**: Points to the 'File' menu in the top toolbar.
- 7. SAVE PROJECT**: Points to the 'Save Project' option in the 'File' menu.
- 8. Generate Code**: Points to the 'Generate Code' option in the 'File' menu.
- 9. Switch to Keil Window**: Points to the 'Switch to Keil Window' option in the 'File' menu.



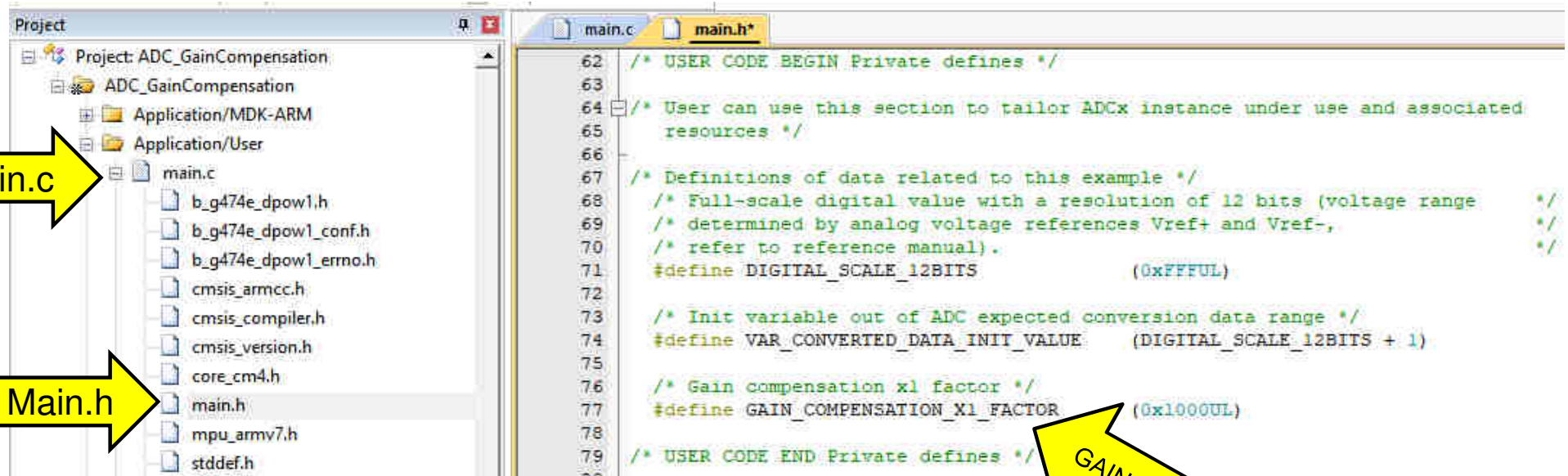
File Changed? Yes & Yes



Since Keil was left open, it understands the Cube made some changes to the underlying files. You do not have to shut down Keil or the Cube, but leave them open. Click yes to each of these windows.



About line 77 in main.h – GAIN_COMPENSATION_X1_FACTOR



Since our previous change was not placed between the USER comment bumpers, we have to make the change again



Paste into line 233 in main.c

The screenshot shows the Keil MDK-ARM IDE interface. The project is named 'ADC_GainCompensation'. The left pane shows the project structure with folders 'Application/MDK-ARM' and 'Application/User', and a file 'main.c'. The main editor displays the code in 'main.c' with line numbers 226 to 238. The code is as follows:

```
226  /* USER CODE END ADC1_Init 1 */
227  /** Common config
228  */
229  hadcl.Instance = ADC1;
230  hadcl.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV2;
231  hadcl.Init.Resolution = ADC_RESOLUTION_12B;
232  hadcl.Init.DataAlign = ADC_DATAALIGN_RIGHT;
233  hadcl.Init.GainCompensation = VDDA_APPLI * GAIN_COMPENSATION_X1_FACTOR / DIGITAL_SCALE_12BITS;
234  hadcl.Init.ScanConvMode = ADC_SCAN_DISABLE;
235  hadcl.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
236  hadcl.Init.LowPowerAutoWait = DISABLE;
237  hadcl.Init.ContinuousConvMode = DISABLE;
238  hadcl.Init.NbrOfConversion = 1;
```

A yellow arrow points to line 233, indicating where the code should be pasted.



Paste into line 59-61 in main.c

```
54  /* USER CODE BEGIN PV */
55  /* Private variables -----
56  /* set to 1 after User push-button interrupt */
57  __IO uint32_t UserButtonStatus = 0;
58
59  uint16_t  Display_DAC1DOR1;
60  uint16_t  Display_DAC3DOR2;
61  uint16_t  Display_ADC3IN1;
62  uint32_t *ts_cal1_p = (uint32_t*)0x1FFF75A8;
63  uint32_t *ts_cal2_p = (uint32_t*)0x1FFF75CA;
64  uint16_t TS_CAL1;
65  uint16_t TS_CAL2;
66
```




Paste into line 62-66 in main.c

```
54  /* USER CODE BEGIN PV */
55  /* Private variables -----
56  /* set to 1 after User push-button interrupt */
57  __IO uint32_t UserButtonStatus = 0;
58
59  uint16_t  Display_DAC1DOR1;
60  uint16_t  Display_DAC3DOR2;
61  uint16_t  Display_ADC3IN1;
62  uint32_t  *ts_cal1_p = (uint32_t*)0x1FFF75A8;
63  uint32_t  *ts_cal2_p = (uint32_t*)0x1FFF75CA;
64  uint16_t  TS_CAL1;
65  uint16_t  TS_CAL2;
66  uint16_t  Display_TemperatureC;
```



Paste into line 131 in main.c

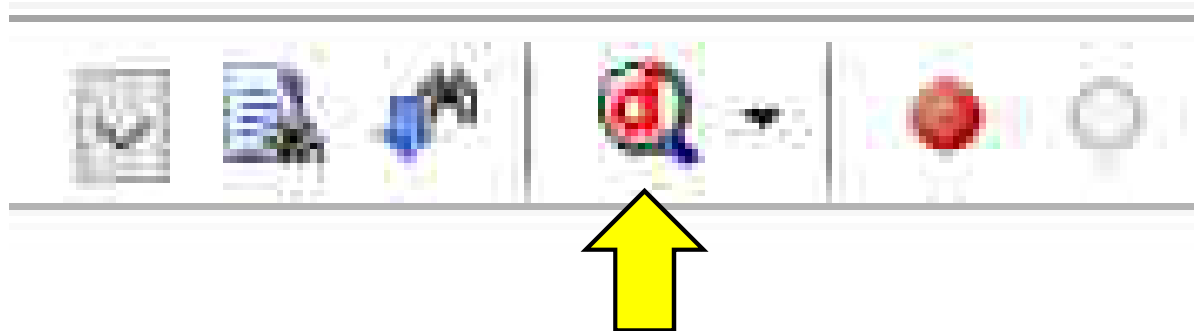
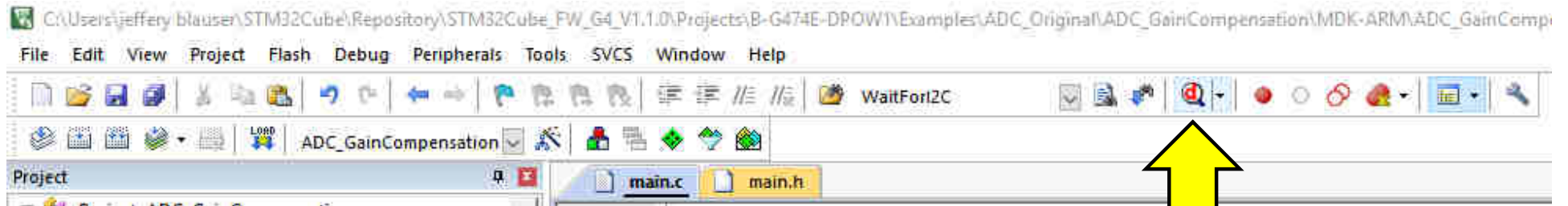
```
126  /* USER CODE BEGIN 2 */
127  /* Initialize the User push-button.
128     It is used for changing the gain */
129  BSP_PB_Init(BUTTON_USER, BUTTON_MODE_EXTI);
130
131  /* Establish the Temperature Sensor Calibration Values */
132  TS_CAL1 = *ts_cal1_p & 0xFFFF; //Read only the two LSBs |
133  TS_CAL2 = *ts_cal2_p & 0xFFFF; //Read only the two LSBs
134
```



Keil MDK-ARM

91

Debugger

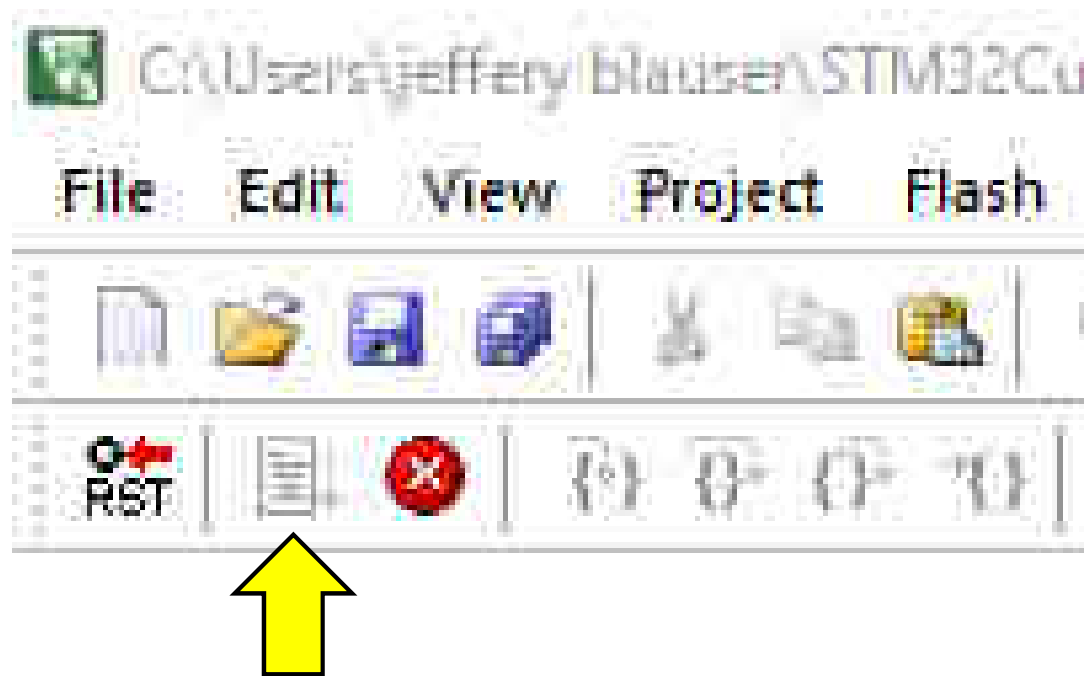




Keil MDK-ARM

92

RUN





VIEW→System Viewer→ADC→ADC1, then ADC1_DR = 0x02F5

1. VIEW

3. ADC

4. ADC1

2. S_V



The screenshot shows the Keil MDK-ARM IDE interface. The 'View' menu is open, and the 'System Viewer' option is highlighted. The 'System Viewer' window is open, showing the 'Core Peripherals' list. The 'ADC' peripheral is selected, and the 'ADC1' sub-item is checked. The 'ADC1' window is open, displaying a table of properties and values.

Property	Value
ISR	0x0000000B
IER	0x00000010
CR	0x10000005
CFGR	0x80001523
CFGR2	0x00010000
SMPR1	0
SMPR2	0x001C0000
TR1	0x0FFF0000
TR2	0x00FF0000
TR3	0x00FF0000
SQR1	0x00000400
SQR2	0
SQR3	0
SQR4	0
DR	0x000002F5
JSQR	0

ADC1_DR=0x02F5



Internal Temperature Sensor

94

Temperature Calculation –RM0440, section 20.4.31

$$Temperature(^{\circ}C) = \frac{TS_{Cal2}Temp - TS_{Cal1}Temp}{TS_{Cal2} - TS_{Cal1}} \times (TS_Data - TS_Cal1) + 30^{\circ}C$$

Table 5. Temperature sensor calibration values

Calibration value name	Description	Memory address
TS_CAL1	TS ADC raw data acquired at a temperature of 30 °C (± 5 °C), V _{DDA} = V _{REF+} = 3.0 V (± 10 mV)	0x1FFF 75A8 - 0x1FFF 75A9
TS_CAL2	TS ADC raw data acquired at a temperature of 110 °C (± 5 °C), V _{DDA} = V _{REF+} = 3.0 V (± 10 mV)	0x1FFF 75CA - 0x1FFF 75CB

$$Temperature(^{\circ}C) = \frac{110 - 30}{1373 - 795} \times (757 - 795) + 30 = 24.7^{\circ}C$$



Internal Temperature Sensor

95

New Gain Factor

$$\text{Temperature } (^{\circ}\text{C}) = \frac{110-30}{1373-795} \times (757-795) + 30 = 24.7^{\circ}\text{C}$$

Our new Gain factor is $\frac{110-30}{1373-795} = 0.238$

We know that the ratio for Gain Compensation is 0-3.999756 for 0-16383 span.

$$\frac{0.238}{3.99756} = \frac{GCompCoeff}{16383}$$

$$GCompCoeff = 975 = 0x03CF$$



Main.h - FILE→SAVE→BUILD

```
75  
76 /* Gain compensation x1 factor */  
77 #define GAIN_COMPENSATION_X1_FACTOR (0x03CFUL)  
78
```



FROM:
0x1000UL

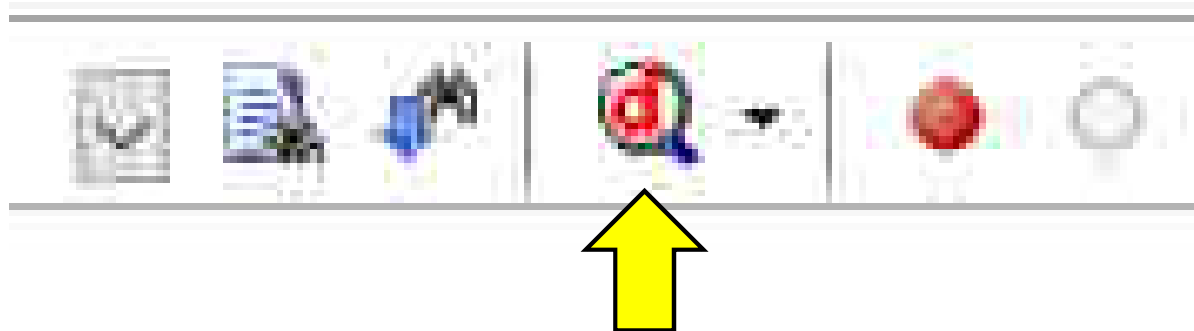
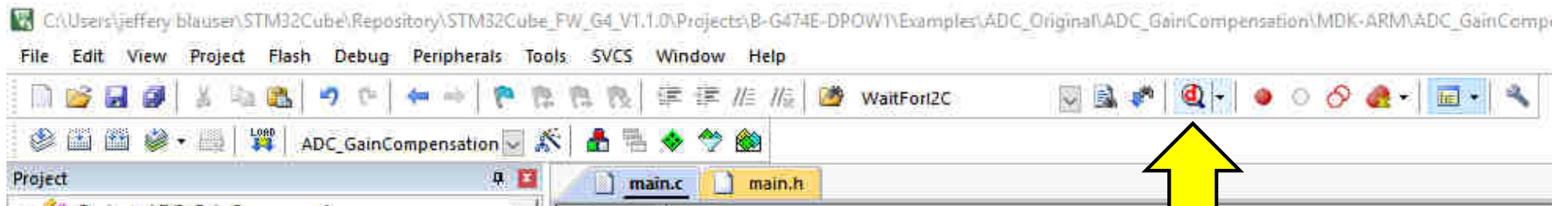
TO:
0x03CFUL



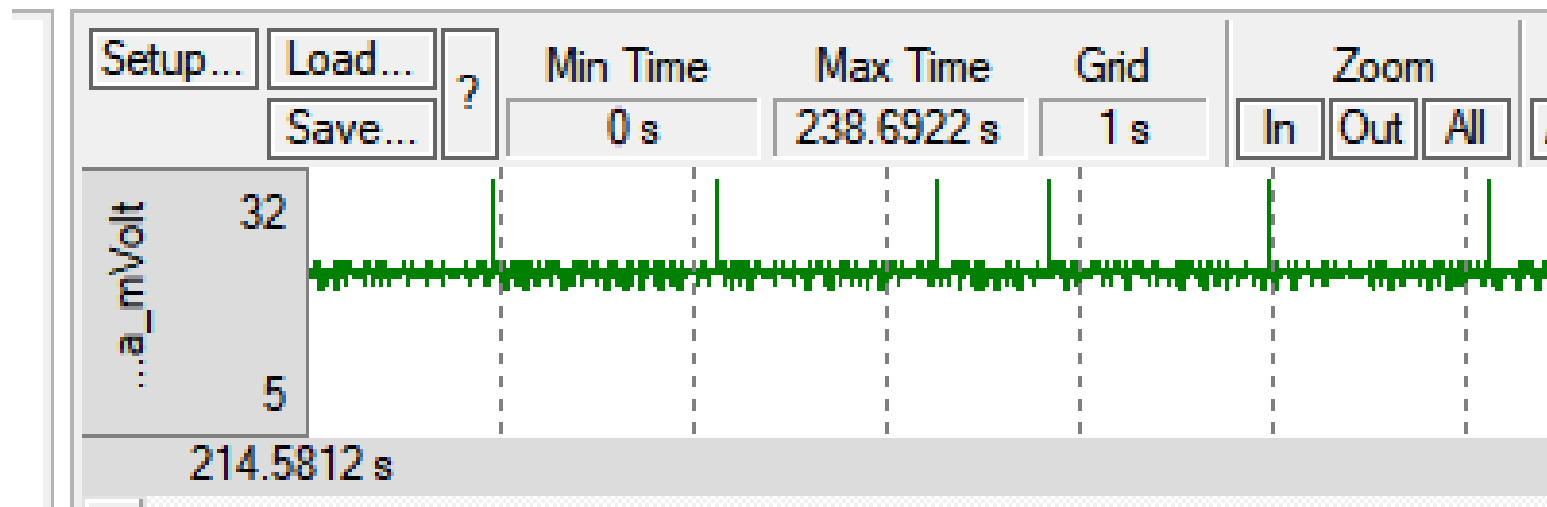
Keil MDK-ARM

97

Debugger



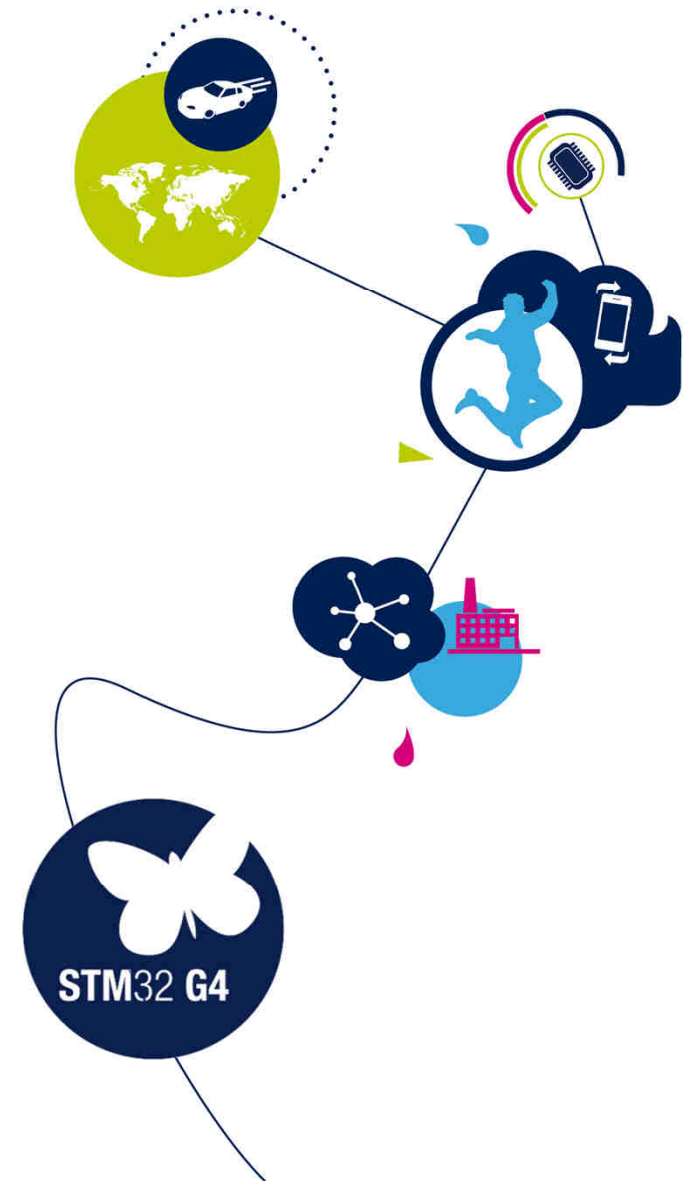
Logic Analyzer Output



Results are in the mid-20°C with just a bit of noise

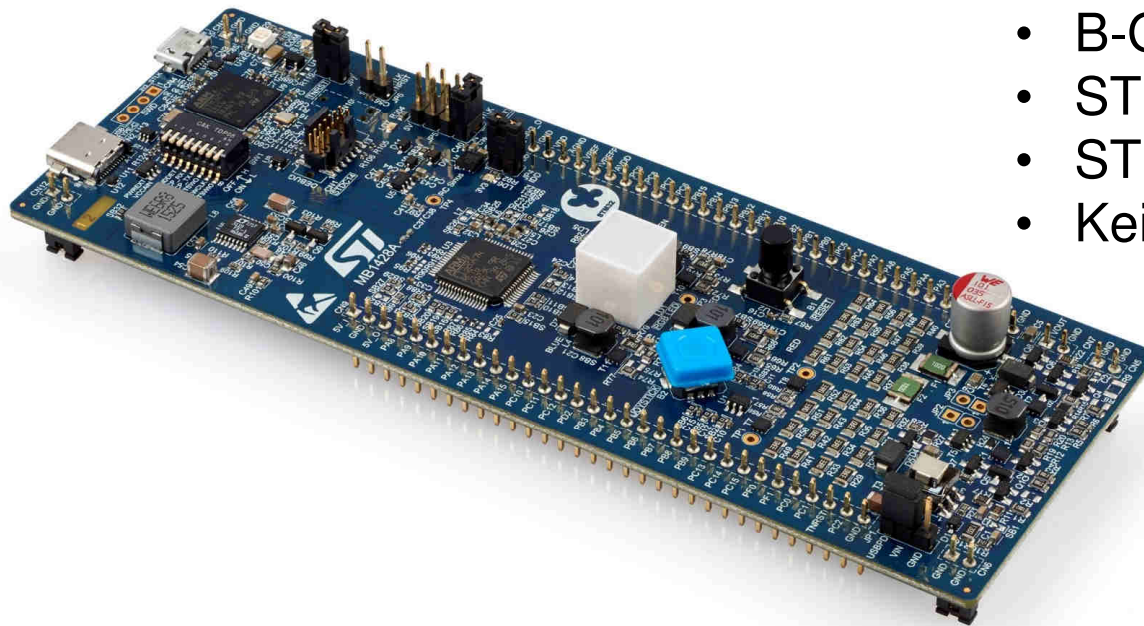
STM32G4 mixed-signal MCU workshop

*Hands On:
DAC Signal Generation*





Systems Check



- USB cable
- B-G474E-DPOW1 Discovery Board
- STM32G4 Family Cube Library
- STM32CubeMx
- Keil MDK-ARM



Digital to Analog Converter

101

G4 DAC

- 3 external outputs:
 - DAC1-OUT1 and OUT2, 1Msps
 - DAC2-OUT1, 1Msps
- 4 internal outputs:
 - DAC3 and DAC4
 - OUT1 and OUT2, 15Msps
- Double data DMA
- Sample & Hold
- Left or Right data alignment
- Signed or unsigned data input format {Q1.15, Q1.11, Q1.7}

Internally connected to:

- Voltage reference
- External Triggers
- Comparator inputs
- Op Amp inputs
- Multiple Timers (+ HRTimer)
- DMA



Digital to Analog Converter

102

G4 DAC Sample and Hold

DAC conversion during the “sample and hold” mode has three phases:

1. Sampling phase: the “sample and hold” element is charged with the desired voltage.
2. Holding phase: the DAC output is tri-stated (High-Z) to maintain the “sample and hold” element’s stored electrical charge.
3. Refresh phase: due to leakage coming from several sources, a refresh phase is essential to maintain the output voltage at the desired value (+/-LSB).



Digital to Analog Converter

103

Dual Conversion from DMA

We will use the DAC dual channel mode with DMA to generate signals on both DAC channels at the same time.

DAC conversions are made with a timer trigger and data fed from DMA.

The timer is configured to trigger the DAC channels at about 1kHz.

DMA is feeding both channels at the time.

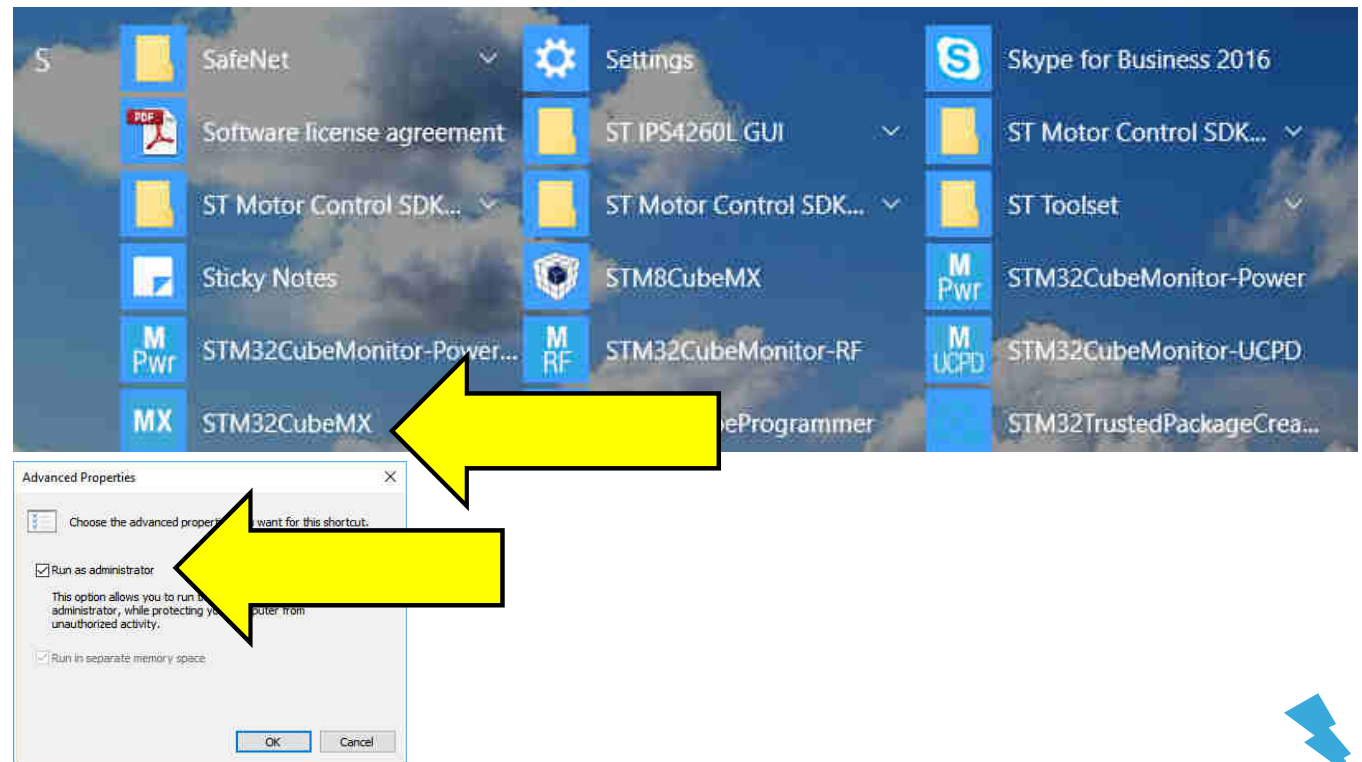
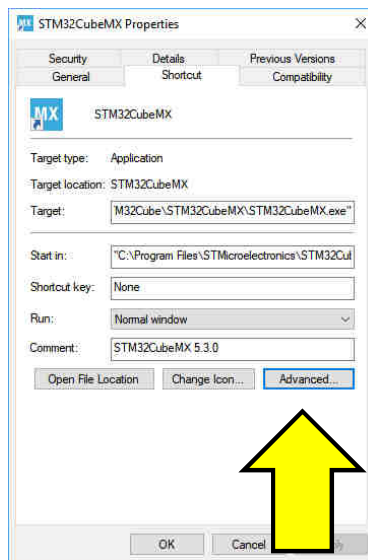
Channel 1 pattern is $\sim 0V / \sim 1.65V / \sim 3.3V / \sim 1.65V$

Channel 2 pattern is $\sim 1.65V / \sim 3.3V / \sim 1.65V / \sim 0V$

We will use the built in Logic Analyzer today instead of an o'scope on the DACs



Start up





Splash Page – G4 Library Installed?

The screenshot displays the STM32CubeMX software interface. On the left, the 'Embedded Software Packages Manager' window is open, showing a table of installed and available software packages. A yellow arrow points to the 'STM32G4' section of the table. The main splash page on the right features a 'Manage software installations' section with a 'CHECK FOR UPDATES' button and an 'INSTALL / REMOVE' button. A large yellow arrow points from the 'INSTALL / REMOVE' button to the right, with the text 'Install/Remove' written inside it. Below the splash page, there is a section for the 'New multicore STM32MP1 Series for Industrial and IoT applications' featuring an STM32MP1 microcontroller and the OpenSTLinux Distribution logo.

Description	Installation	Available Version
STM32F7		
STM32G0		
STM32G4		
STM32Cube MCU Package for STM32G4 Series	1.1.0	1.1.0
STM32Cube MCU Package for STM32G4 Series	1.0.1	1.0.1

Details

From Local From Url Refresh Install Now Remove Now Close

Manage software installations

Check for STM32CubeMX and embedded software packs...

CHECK FOR UPDATES

Install or remove embedded software packages

INSTALL / REMOVE

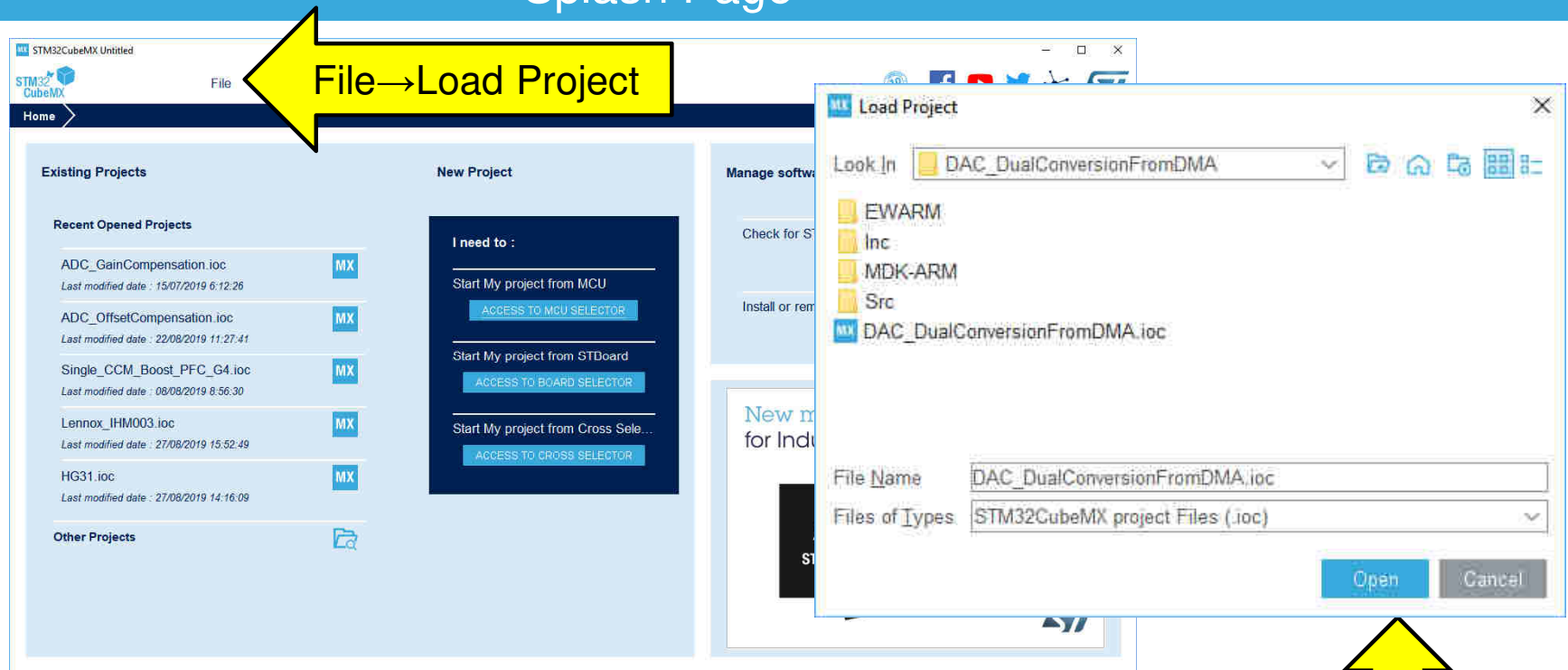
New multicore STM32MP1 Series for Industrial and IoT applications

STM32MP1

OpenSTLinux Distribution



Splash Page



C:\...\STM32Cube\Repository\STM32Cube_FW_G4_V1.1.0\Projects\B-G474E-DPOW1\Examples\DAC\DAC_DualConversionFromDMA



Graphic Design Representation





DAC1, Timer 2

The screenshot shows the STM32CubeMX DAC1 configuration window for an STM32G474RETx. The left sidebar lists various components, with DAC1 selected under the Analog category. The main panel displays the DAC1 Mode and Configuration settings, including Mode, OUT1 mode, OUT2 mode, and External Trigger. The Configuration tab is active, showing DAC Out1 and DAC Out2 settings. Two yellow arrows point to the 'Timer 2 Trigger Out event' field in the DAC Out1 and DAC Out2 settings, both labeled 'Timer 2 Trigger Out'. The right side of the window shows a pinout view of the STM32G474RETx chip, with pins PA15 and PA14 highlighted in green, corresponding to the DAC1_OUT1 and DAC1_OUT2 pins.

DAC1

Timer 2 Trigger Out

Timer 2 Trigger Out



Project Manager – Keil MDK-ARM V5

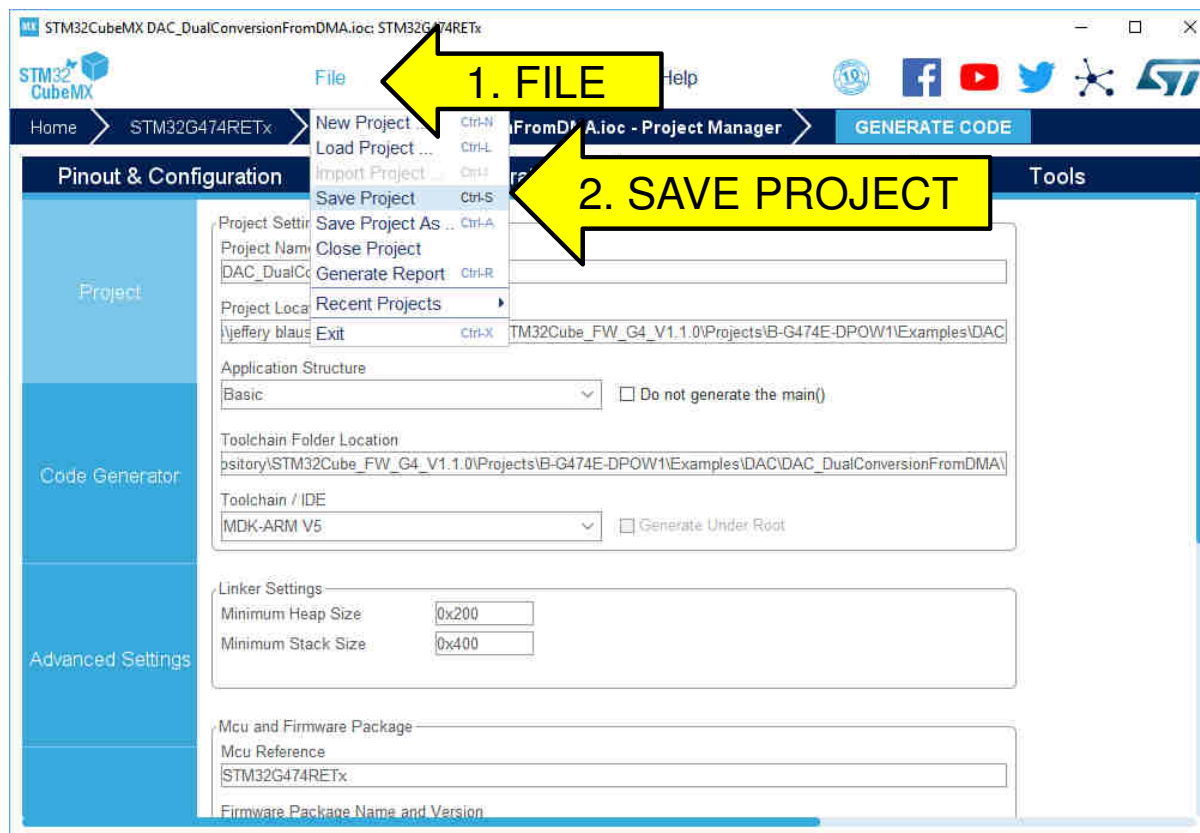
The screenshot displays the STM32CubeMX Project Manager window for a project named "DAC_DualConversionFromDMA.ioc: STM32G474RETx". The interface includes a top menu bar with "File", "Window", and "Help", and a breadcrumb trail: "Home > STM32G474RETx > DAC_DualConversionFromDMA.ioc - Project Manager". A "GENERATE CODE" button is located in the top right. The left sidebar contains four tabs: "Pinout & Configuration", "Clock Configuration", "Project Manager" (which is active), and "Tools". The "Project Manager" tab shows several configuration sections:

- Project Settings:**
 - Project Name: DAC_DualConversionFromDMA
 - Project Location: \\jeffery blausen\STM32Cube\Repository\STM32Cube_FW_G4_V1.1.0\Projects\B-G474E-DPOW1\Examples\DAC_DualConversionFromDMA
 - Application Structure: Basic (dropdown menu) with a checkbox for "Do not generate the main()".
- Code Generator:**
 - Toolchain Folder Location: repository\STM32Cube_FW_G4_V1.1.0\Projects\B-G474E-DPOW1\Examples\DAC_DualConversionFromDMA
 - Toolchain / IDE: MDK-ARM V5 (dropdown menu)
- Advanced Settings:**
 - Linker Settings:
 - Minimum Heap Size: 0x200
 - Minimum Stack Size: 0x400
 - Mcu and Firmware Package:
 - Mcu Reference: STM32G474RETx
 - Firmware Package Name and Version: (empty field)

Two yellow arrows are overlaid on the image: one points to the "Project Manager" tab, and the other points to the "MDK-ARM V5" option in the "Toolchain / IDE" dropdown menu.

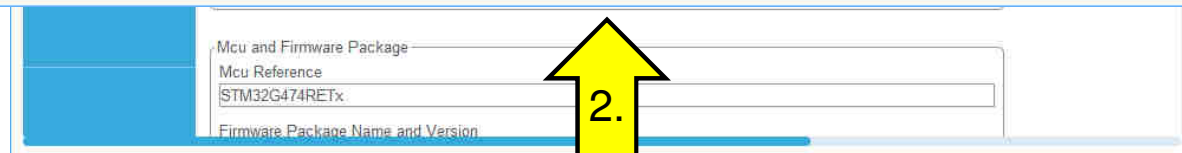
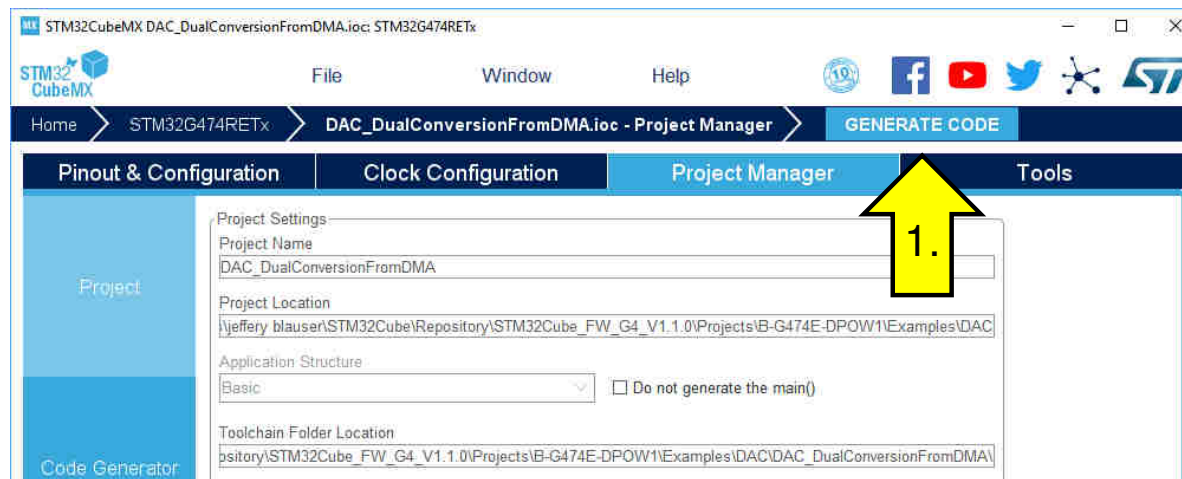


FILE→SAVE PROJECT





Generate Code





Keil Start up page

```
128 if (HAL_DACEx_DualStart_DMA(&hdac1, DAC_CHANNEL_1, wave, sizeof(wave))
129 {
130     /* DAC conversion start error */
131     Error_Handler();
132 }
133
134 BSP_LED_On(LED4);
135 while (1)
136 {
137     /* USER CODE END WHILE */
138
139     /* USER CODE BEGIN 3 */
140 }
141 /* USER CODE END 3 */
142
143
144 /**
145  * @brief System Clock Configuration
146  * @retval None
147  */
148 void SystemClock_Config(void)
149 {
150     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
151     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
152
153     /** Configure the main internal regulator output voltage
154     */
155     HAL_PWREx_ControlVoltageScaling(PWR_REGULATOR_VOLTAGE_SCALE1);
156     /** Initializes the CPU, AHB and APB busses clocks
157     */
158     RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
159     RCC_OscInitStruct.HSIState = RCC_HSI_ON;
160     RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
161     RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
162     RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
163     RCC_OscInitStruct.PLL.PLLM = RCC_PLLM_DIV4;
164     RCC_OscInitStruct.PLL.PLLN = 75;
165     RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
166     RCC_OscInitStruct.PLL.PLLQ = RCC_PLLQ_DIV2;
167     RCC_OscInitStruct.PLL.PLLR = RCC_PLLR_DIV2;
168     if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
169     {
170         Error_Handler();
171     }
172     /** Initializes the CPU, AHB and APB busses clocks
```

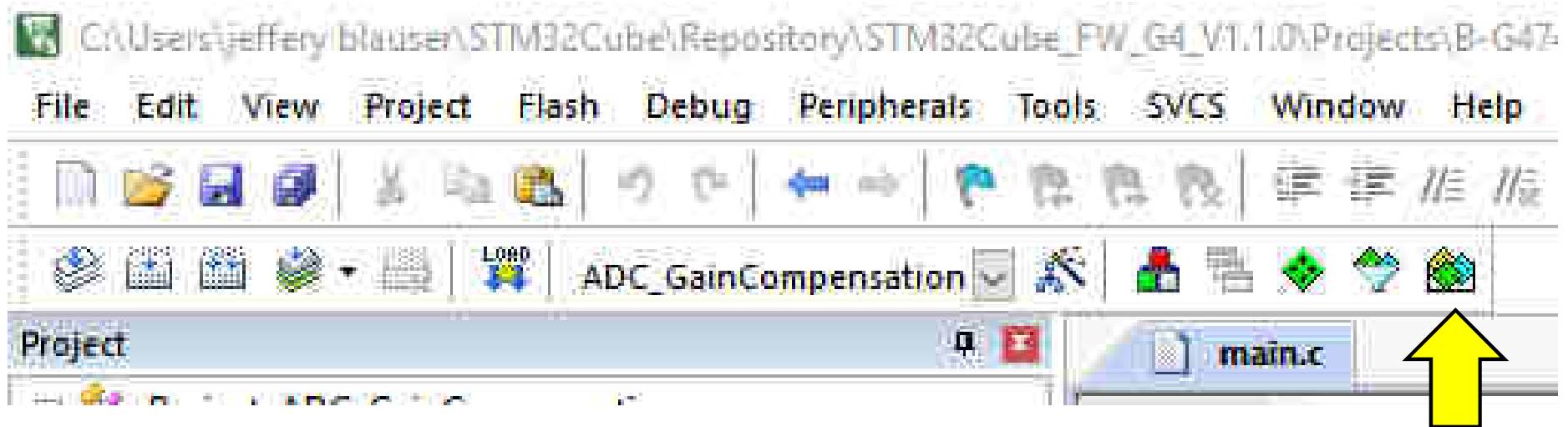




Keil MDK-ARM

113

Check the Device Pack





G4 Device Pack

The screenshot shows the Keil MDK-ARM Pack Installer window. The 'Devices' tab is active, displaying a list of devices. The 'STM32G474' device is highlighted with a yellow arrow. The 'Packs' tab is also visible, showing a list of packs. The 'Keil:STM32G4xx_DFP' pack is highlighted with a yellow arrow. A yellow arrow labeled 'Exit' points to the top right corner of the window. The output window at the bottom shows error messages related to downloading pack descriptions.

Device: STM32G474

Search:

Device

- Silicon Labs (906 Devices)
- Sinowearth (1 Device)
- SONIX (50 Devices)
- STMicroelectronics (1284 Devices)
 - STBlueNRG Series (1 Device)
 - STBlueNRG-1 Series (1 Device)
 - STBlueNRG-2 Series (1 Device)
 - STM32F0 Series (111 Devices)
 - STM32F1 Series (95 Devices)
 - STM32F2 Series (46 Devices)
 - STM32F3 Series (90 Devices)
 - STM32F4 Series (205 Devices)
 - STM32F7 Series (115 Devices)
 - STM32G0 Series (34 Devices)
 - STM32G4 Series (118 Devices)
 - STM32G431 (24 Devices)
 - STM32G441 (8 Devices)
 - STM32G471 (17 Devices)
 - STM32G473 (25 Devices)
 - STM32G474 (25 Devices)
 - STM32G483 (9 Devices)
 - STM32G484 (9 Devices)
 - STM32GBK1 (1 Device)
 - STM32H7 Series (22 Devices)
 - STM32L0 Series (152 Devices)
 - STM32L1 Series (81 Devices)

Summary

Pack

Pack	Action	Description
Device Specific	1 Pack	STM32G4xx_DFP
Generic	40 Packs	
Alibaba:AliOSThings	Install	AliOS Things software pack
ARM:AMP	Install	Software components for inter processor communication (Asymmetric Multi
ARM:CMSIS	Up to date	CMSIS (Cortex Microcontroller Software Interface Standard)
ARM:CMSIS-Driver	Up to date	CMSIS Drivers for external devices
ARM:CMSIS-Driver_Val...	Up to date	CMSIS-Driver Validation
ARM:CMSIS-FreeRTOS	Up to date	Bundle of FreeRTOS for Cortex-M and Cortex-A
ARM:CMSIS-RTOS_Val...	Up to date	CMSIS-RTOS Validation
ARM:mbedClient	Install	ARM mbed Client for Cortex-M devices
ARM:mbedTLS	Install	ARM mbed Cryptographic and SSL/TLS library for Cortex-M devices
ARM:minar	Install	mbed OS Scheduler for Cortex-M devices
ARM:TFM	Install	Trusted Firmware-M (TF-M) is the reference implementation of Arm's Platfor
EmbeddedOffice:Flexi...	Install	Flexible Safety RTOS
Keil:ARM_Compiler	Up to date	Keil ARM Compiler extensions for ARM Compiler 5 and ARM Compiler 6
Keil:IMXRT105x_MWP	Install+	NXP i.MX RT 105x MDK-Middleware examples and CMSIS-Drivers
Keil:Jansson	Install	Jansson is a C library for encoding, decoding and manipulating JSON data
Keil:MDK-Middleware	Up to date	Middleware for Keil MDK-Professional and MDK-Plus
lwIP:lwIP	Install	lwIP is a light-weight implementation of the TCP/IP protocol suite
MDK-Packs:AWS_IoT_...	Install+	SDK for connecting to AWS IoT from a device using embedded C
MDK-Packs:Azure_IoT	Install+	Microsoft Azure IoT SDKs and Libraries
MDK-Packs:cJSON	Install	Ultralightweight JSON parser in ANSI C
MDK-Packs:Google_Io...	Install+	Google Cloud IoT Device Connector
MDK-Packs:IoT_Socket	Install	Simple IP Socket (BSD like)
MDK-Packs:Jsmin	Install	Minimalistic JSON parser/tokenizer in C
MDK-Pack:PaHo MQTT	Install+	Embedded MQTT C/C++ Client Libraries

Output

Cannot download file http://mcuxpresso.nxp.com/cmsis_pack/repo/NXP.LPC54018M_DFP.pdsc: Illegal MIME type 'text/html'

Cannot download file http://mcuxpresso.nxp.com/cmsis_pack/repo/NXP.LPC54018_DFP.pdsc: Illegal MIME type 'text/html'

Cannot download file http://mcuxpresso.nxp.com/cmsis_pack/repo/NXP.LPC54101_DFP.pdsc: Illegal MIME type 'text/html'

Action (2 left): Update Pack descriptions, download http://mcuxpresso.nxp.com/cmsis_pack/repo/NXP.LPC54102_DFP.pdsc

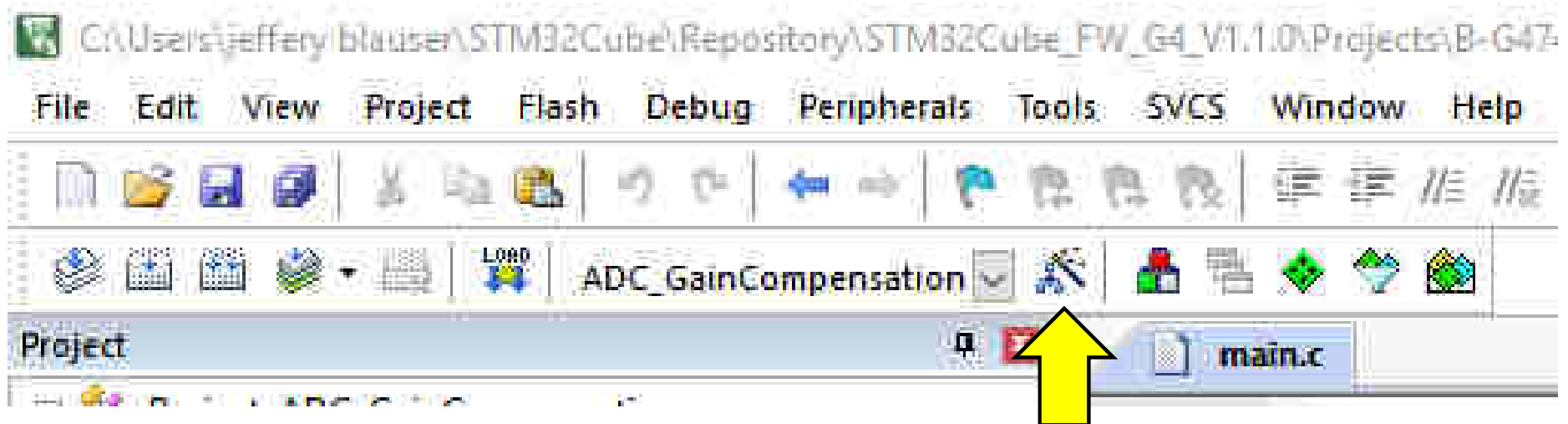
0% ONLINE



Keil MDK-ARM

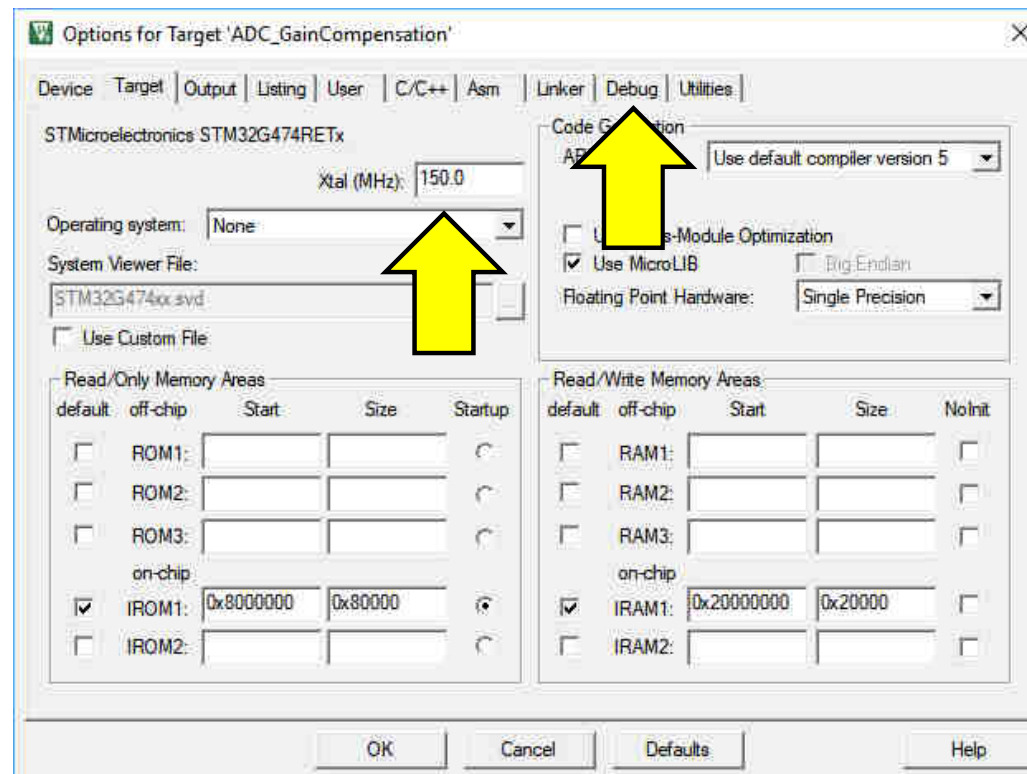
115

Check the Options



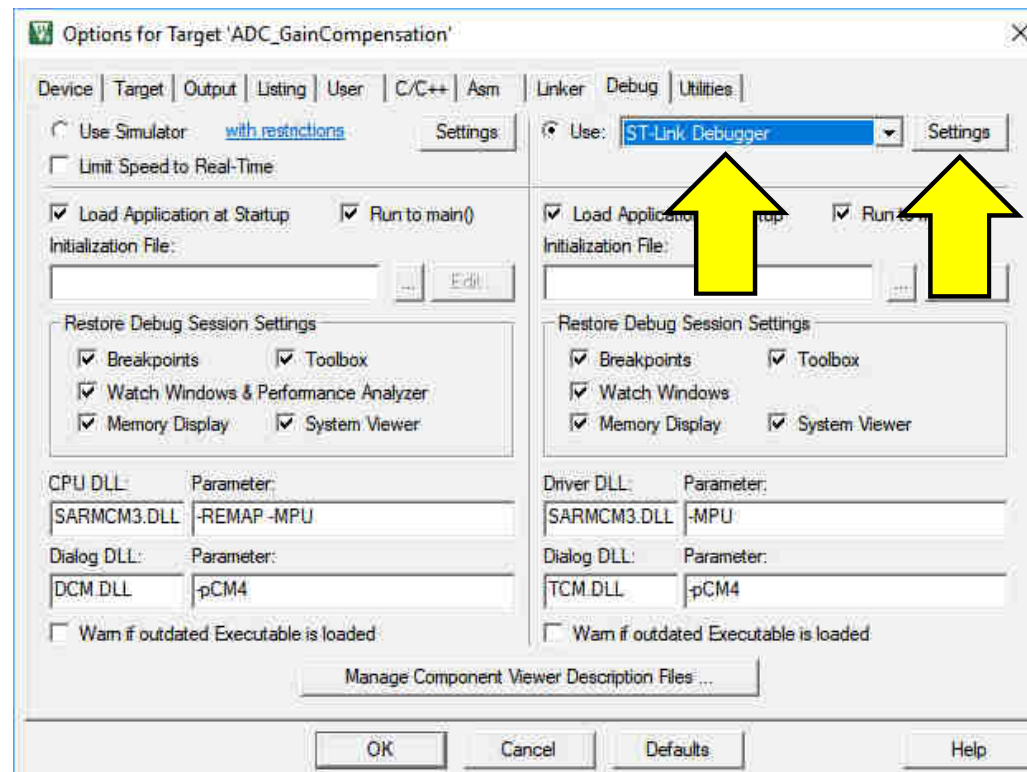


Clock speed and Debug



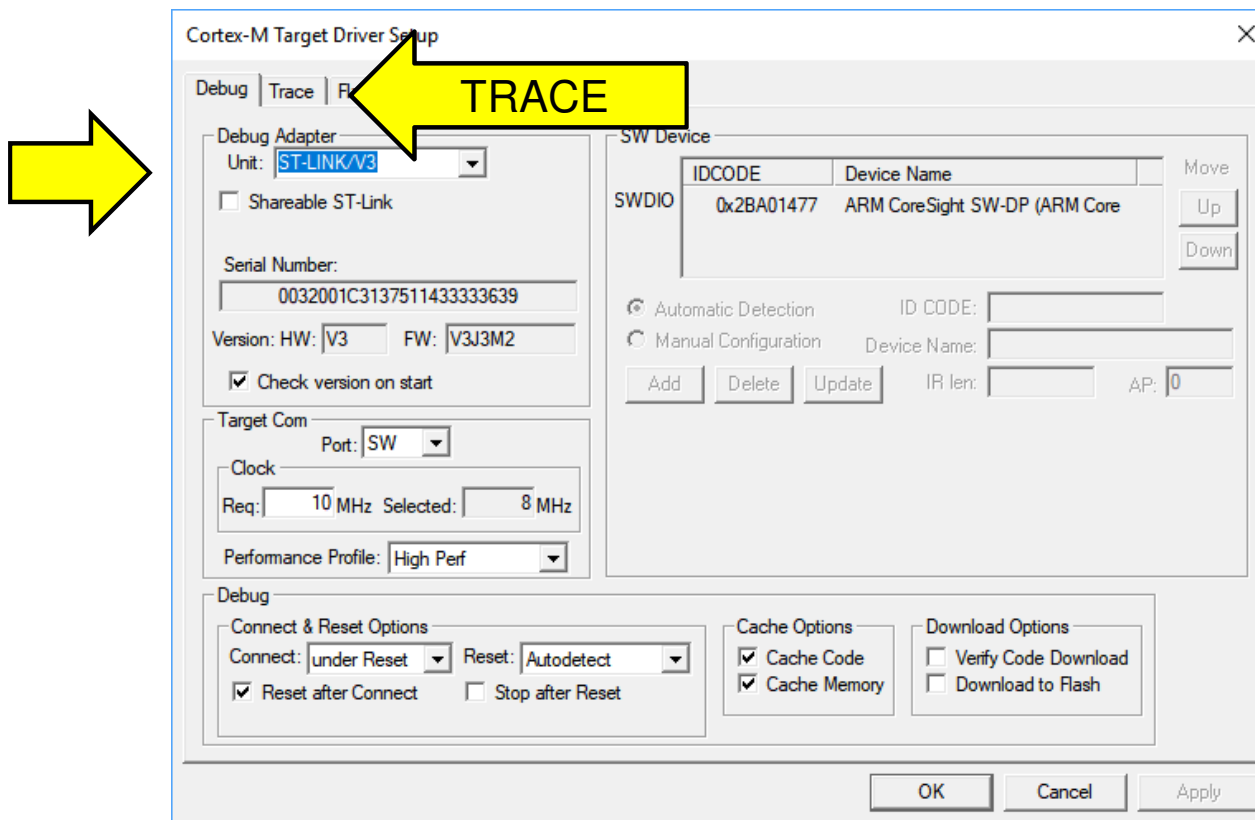


ST-Link Debugger - Settings





ST-Link/V3 & TRACE





Core Clock & TRACE Enable (Before)

The screenshot shows the 'Cortex-M Target Driver Setup' dialog box with the 'Trace' tab selected. The following annotations highlight specific settings:

- 1. 150**: Points to the 'Core Clock' field, which is set to 150.000000 MHz.
- 2. TRACE Enable**: Points to the 'Trace Enable' checkbox, which is checked.
- 3. Uncheck EXCTRC**: Points to the 'EXCTRC: Exception Tracing' checkbox in the 'Trace Events' section, which is checked.
- 4. Uncheck all but 15**: Points to the 'Port 15..8' checkbox in the 'ITM Stimulus Ports' section, which is checked.
- 5. OK**: Points to the 'OK' button at the bottom right.

The dialog box contains the following sections and settings:

- Debug** | **Trace** | Flash Download | Pack
- Core Clock**: 150.000000 MHz
- Trace Port**: Serial Wire Output - UART/NRZ
- SWO Clock Prescaler**: 13
- SWO Clock**: 11.538461 MHz
- Autodetect**: ☒
- Trace Events**:
 - ☒ EXCTRC: Exception Tracing
 - ☐ CPI: Cycles per Instruction
 - ☐ EXC: Exception overhead
 - ☐ SLEEP: Sleep Cycles
 - ☐ LSU: Load Store Unit Cycles
 - ☐ FOLD: Folded Instructions
- ITM Stimulus Ports**:

31	Port	24	23	Port	16	15	Port	8	7	Port	0	
☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	
Port 31..24				☒	Port 23..16				☐	Port 15..8		☐
Port 31..24				☒	Port 23..16				☐	Port 15..8		☐
Port 31..24				☒	Port 23..16				☐	Port 15..8		☐
Port 31..24				☒	Port 23..16				☐	Port 15..8		☐
- Advanced settings**:
 - ☐ Ignore packets with no SYNC
 - ☐ Overwrite CYCCNT



Core Clock & TRACE Enable (After)

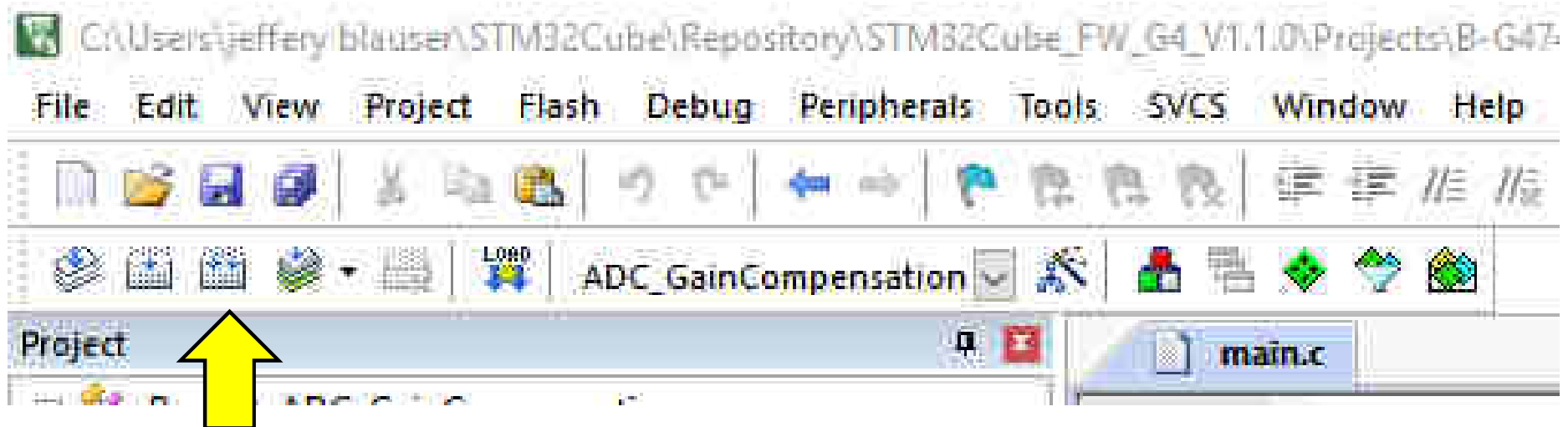
The screenshot shows the 'Cortex-M Target Driver Setup' dialog box with the 'Trace' tab selected. The following settings are visible:

- Core Clock:** 150.000000 MHz (Annotated with '1.150')
- Trace Enable:** ☒ (Annotated with '2. TRACE Enable')
- Trace Port:** Serial Wire Output - UART/NRZ
- SWO Clock Prescaler:** 13
- Autodetect:** ☒
- SWO Clock:** 11.538461 MHz
- Timestamps:** ☒ Enable, Prescaler: 1
- PC Sampling:** ☒ Periodic, Period: 109.227 us, ☐ on Data R/W Sample
- Trace Events:**
 - ☐ CPI: Cycles per Instruction
 - ☐ EXC: Exception overhead
 - ☐ SLEEP: Sleep Cycles
 - ☐ LSU: Load Store Unit Cycles
 - ☐ FOLD: Folded Instructions
 - ☐ EXCTRC: Exception Tracing (Annotated with '3. Uncheck EXCTRC')
- ITM Stimulus Ports:**

31	Port	24	23	Port	16	15	Port	8	7	Port	0
Enable:	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐	☐
Privilege:	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐	☐
	Port 31..24	☒	Port 23..16	☐	Port 15..8	☒	Port 7..0	☐			
- Advanced settings:**
 - ☐ Ignore packets with no SYNC
 - ☐ Overwrite CYCCNT
- Buttons:** OK, Cancel, Apply (Annotated with '5. OK' pointing to the OK button)



Build

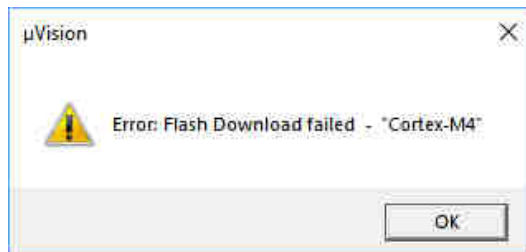
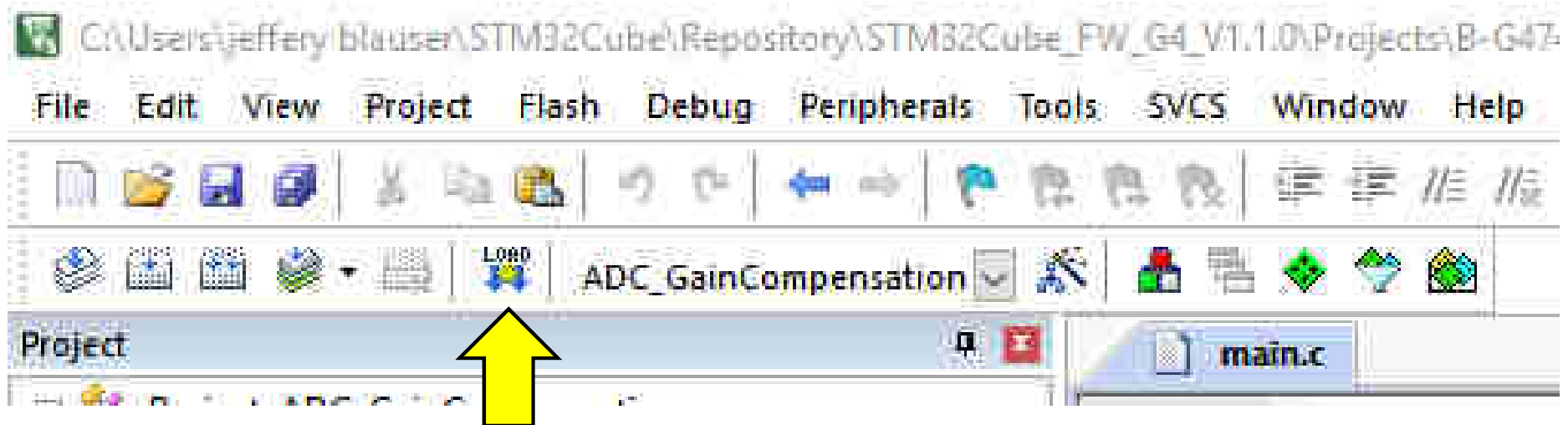


Build Output

```
Program Size: Code=6572 RO-data=576 RW-data=92 ZI-data=1204
FromELF: creating hex file...
DAC_DualConversionFromDMA\DAC_DualConversionFromDMA.hex: Warning: Q9931W: Your license for feature Keil will expire in 30 days
Finished: 0 information, 1 warning and 0 error messages.
"DAC_DualConversionFromDMA\DAC_DualConversionFromDMA.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:01:38
```



Load



Build Output

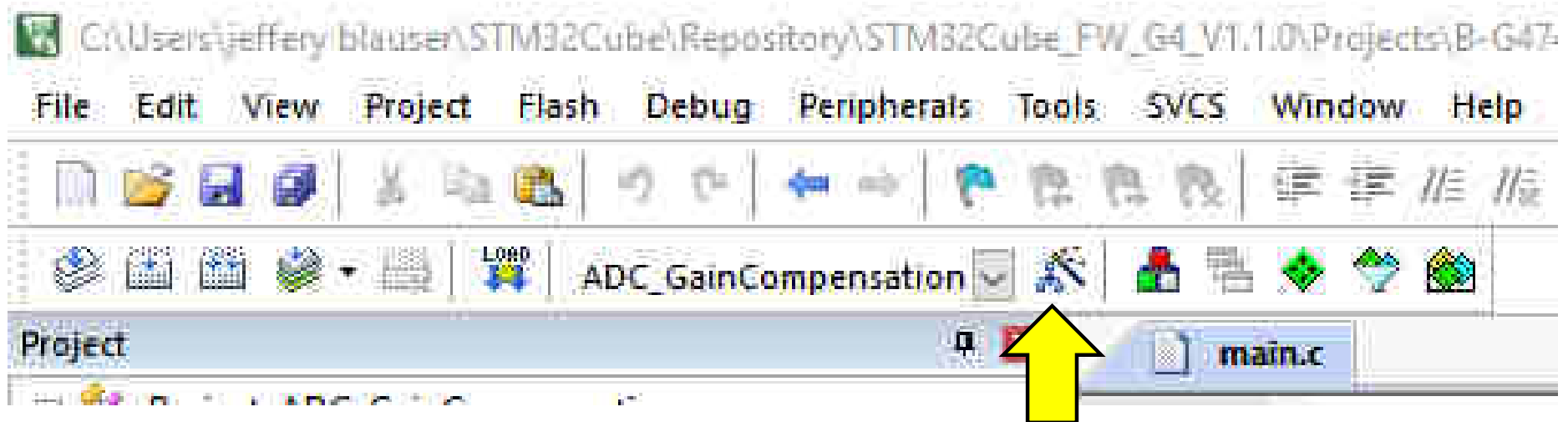
```
Contents mismatch at: 08001860H (Flash=38H Required=C5H) !
Contents mismatch at: 08001861H (Flash=46H Required=F8H) !
Contents mismatch at: 08001862H (Flash=FEH Required=88H) !
Contents mismatch at: 08001863H (Flash=F7H Required=00H) !
Contents mismatch at: 08001864H (Flash=B9H Required=60H) !
Contents mismatch at: 08001865H (Flash=FCH Required=6CH) !
Too many errors to display !
Error: Flash Download failed - "Cortex-M4"
```



Keil MDK-ARM

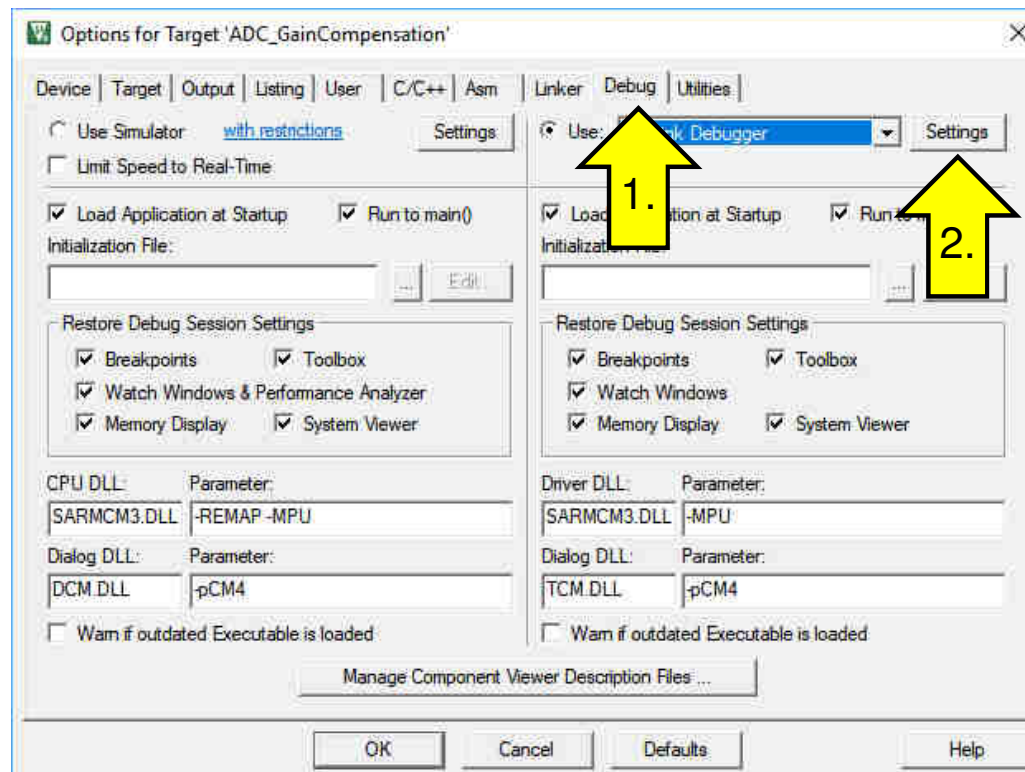
123

Options



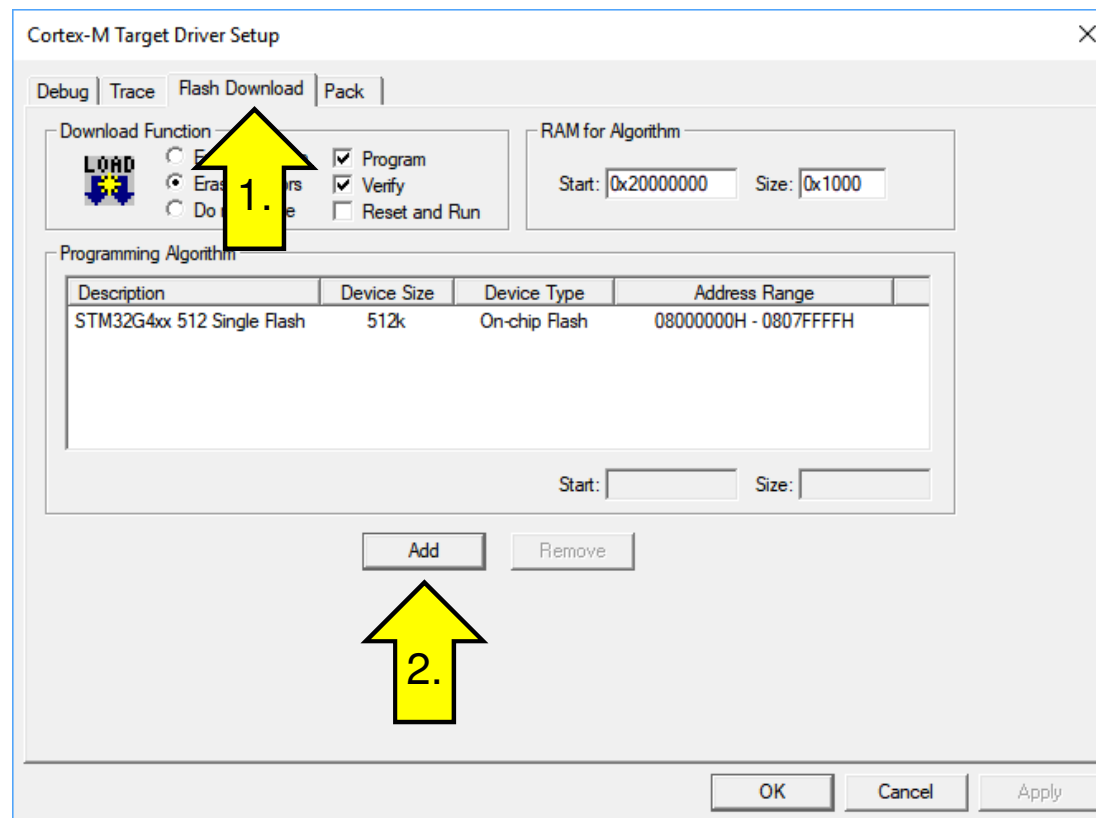


ST-Link Debugger - Settings



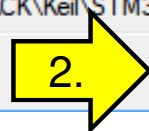
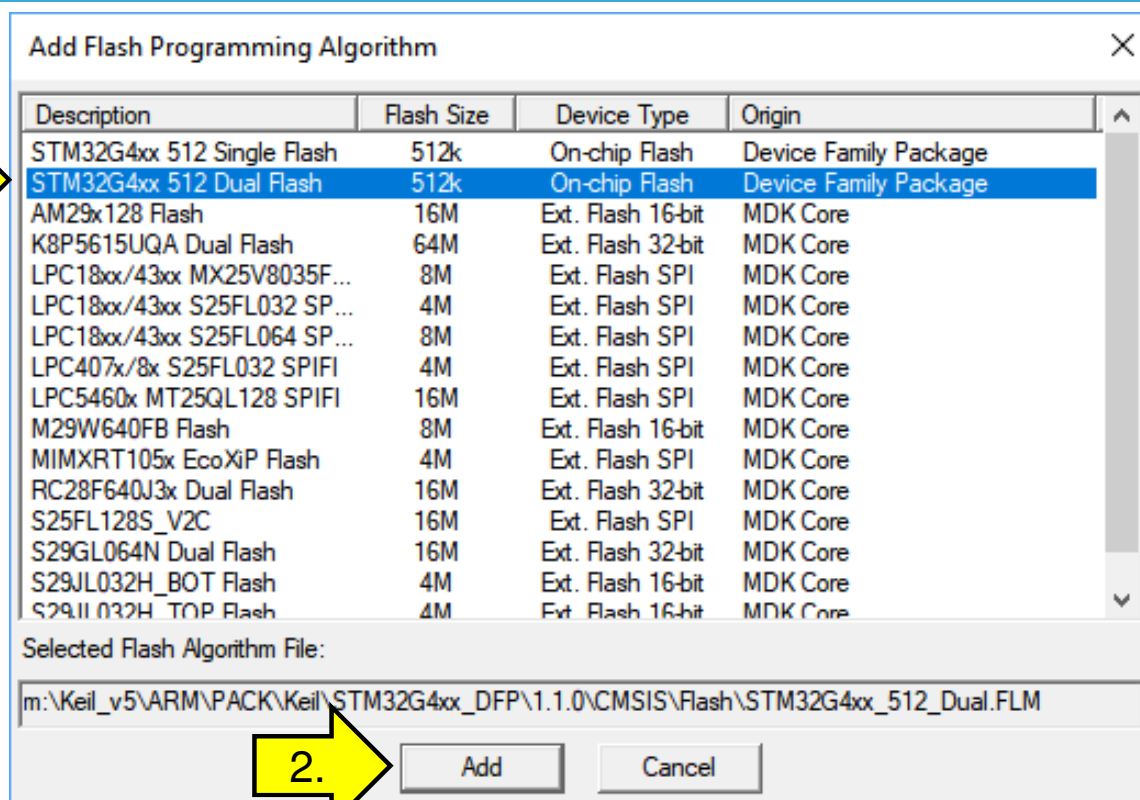


Flash Download → Add

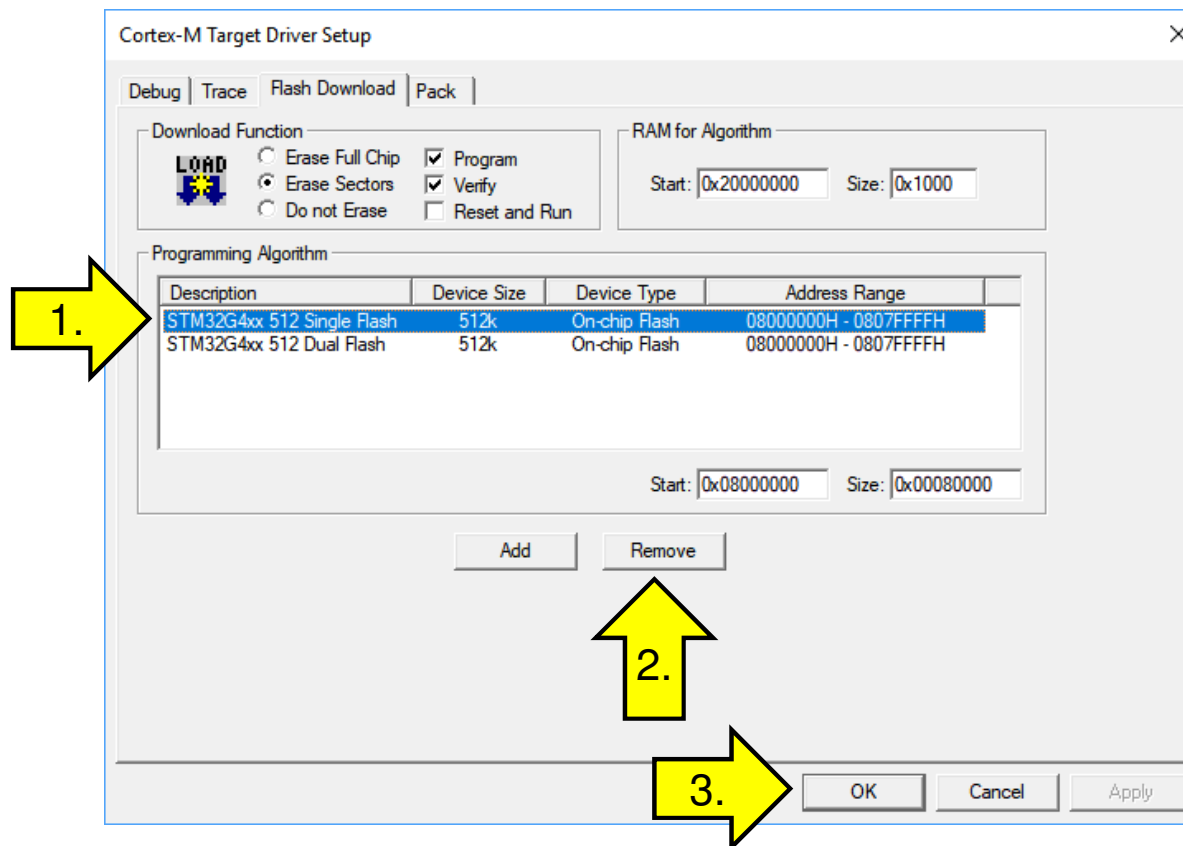




STM32G4xx 512 Dual Flash



Remove Single Flash





Place 2 new variables for the Analyzer – main.c about line 56

```
51  /* USER CODE BEGIN PV */
52  /* Private variables -----
53
54  uint32_t wave[4] = {0x07FF0
55
56  uint16_t Display_DAC1DOR1;
57  uint16_t Display_DAC1DOR2;
58
59  /* USER CODE END PV */
```

See c:\2019_STM32G4_Workshop\HandsOn_Labs\DAC
For the file CodeLinesForMain_DAC.txt for cut and paste.



Execute 3 new lines for the Analyzer – main.c about line 143

```
138 while (1)
139 {
140     /* USER CODE END WHILE */
141
142     /* USER CODE BEGIN 3 */
143     Display_DAC1DOR1 = DAC1->DOR1;
144     Display_DAC1DOR2 = DAC1->DOR2;
145     HAL_Delay(25);
146 }
```

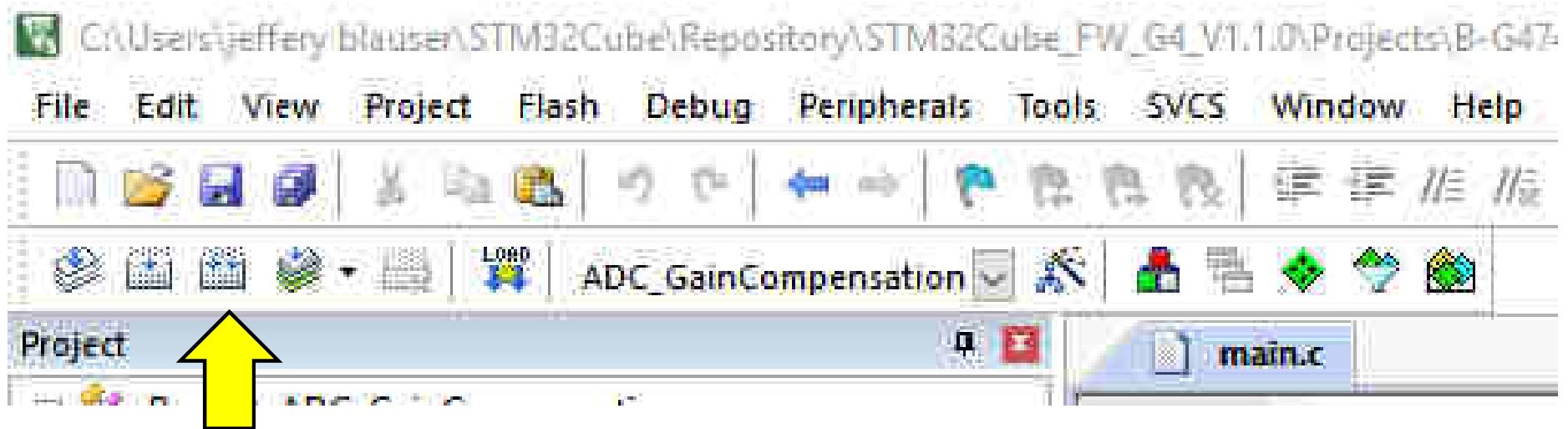
See c:\2019_STM32G4_Workshop\HandsOn_Labs\DAC
For the file CodeLinesForMain_DAC.txt for cut and paste.



Keil MDK-ARM

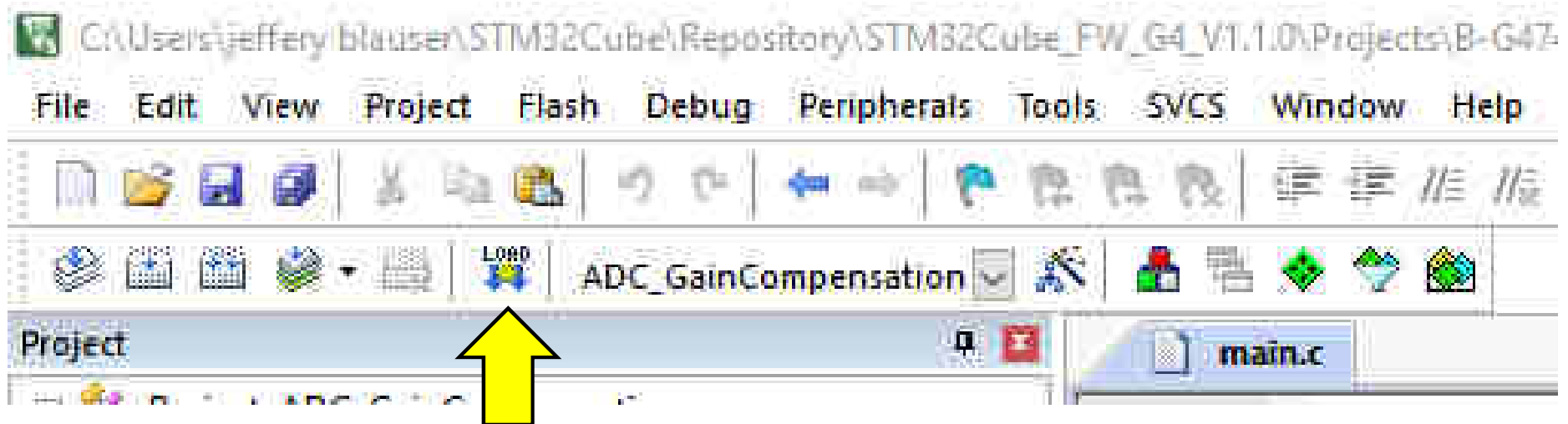
130

FILE→SAVE→Build





Load



Build Output

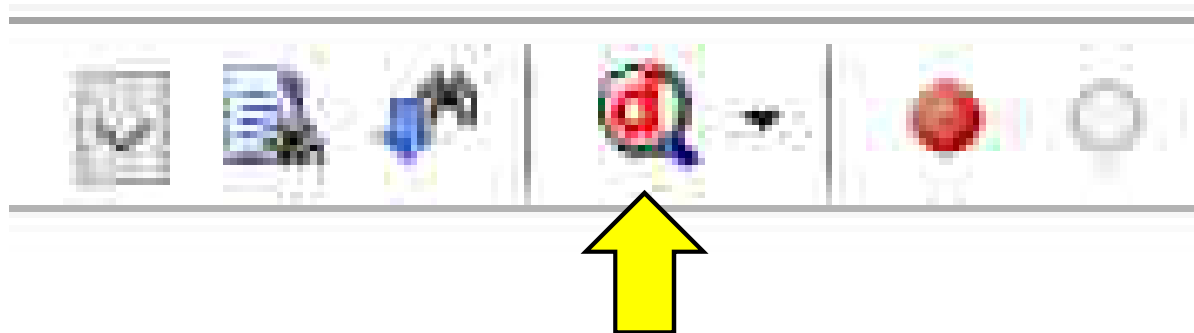
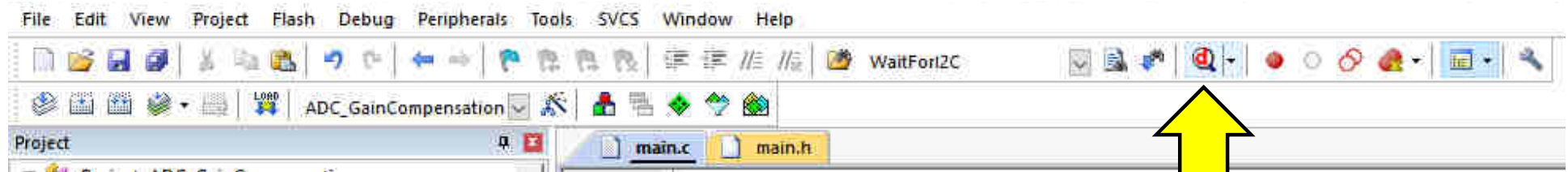
```
Build Time Elapsed: 00:01:38
Load "DAC_DualConversionFromDMA\\DAC_DualConversionFromDMA.axf"
Erase Done.
Programming Done.
Verify OK.
Flash Load finished at 14:51:03
```



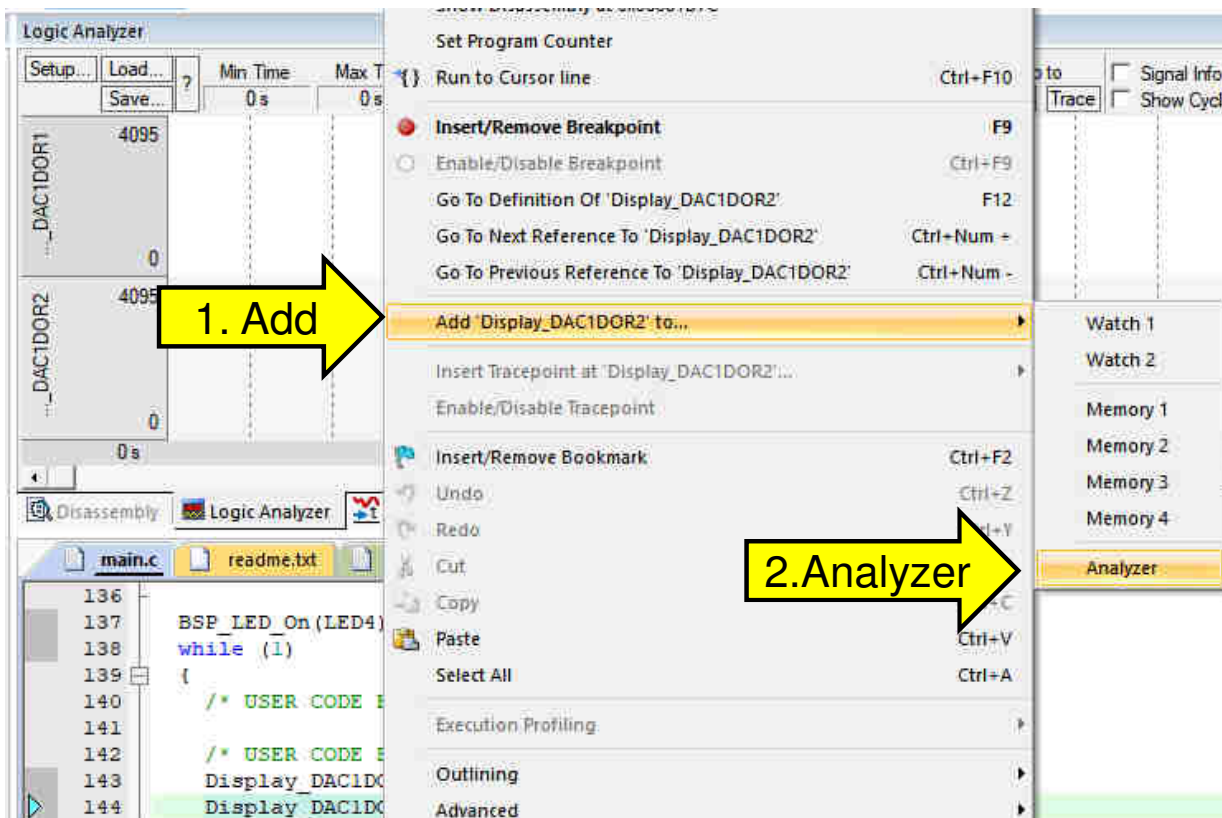
Keil MDK-ARM

132

Debugger



Add to Analyzer Display_DAC1DOR1 & 2



The screenshot shows the Keil MDK-ARM Logic Analyzer interface. The main window displays the Logic Analyzer setup, including a table of variables to be analyzed. The variables listed are `Display_DAC1DOR1` and `Display_DAC1DOR2`, both with a value of 0. A yellow arrow labeled "1. Add" points to the "Add" button in the Logic Analyzer window.

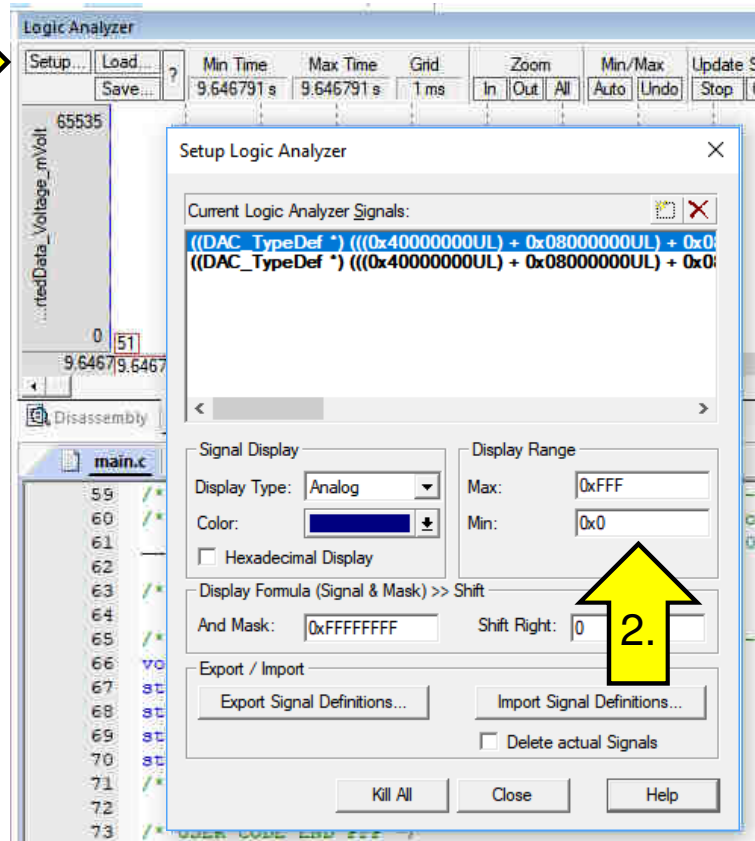
The Logic Analyzer window includes a "Setup..." button, a "Load..." button, a "Save..." button, and a "Min Time" field set to 0s. The "Max Time" field is also set to 0s. The "Add" button is highlighted with a yellow arrow labeled "1. Add".

The Logic Analyzer window also displays a list of variables to be analyzed, including `Display_DAC1DOR1` and `Display_DAC1DOR2`. The values for these variables are shown as 0. A yellow arrow labeled "2. Analyzer" points to the "Analyzer" button in the Logic Analyzer window.

The Logic Analyzer window also displays a list of variables to be analyzed, including `Display_DAC1DOR1` and `Display_DAC1DOR2`. The values for these variables are shown as 0. A yellow arrow labeled "2. Analyzer" points to the "Analyzer" button in the Logic Analyzer window.

Setup – Min=0x0, Max = 0xFFFF

1. Setup

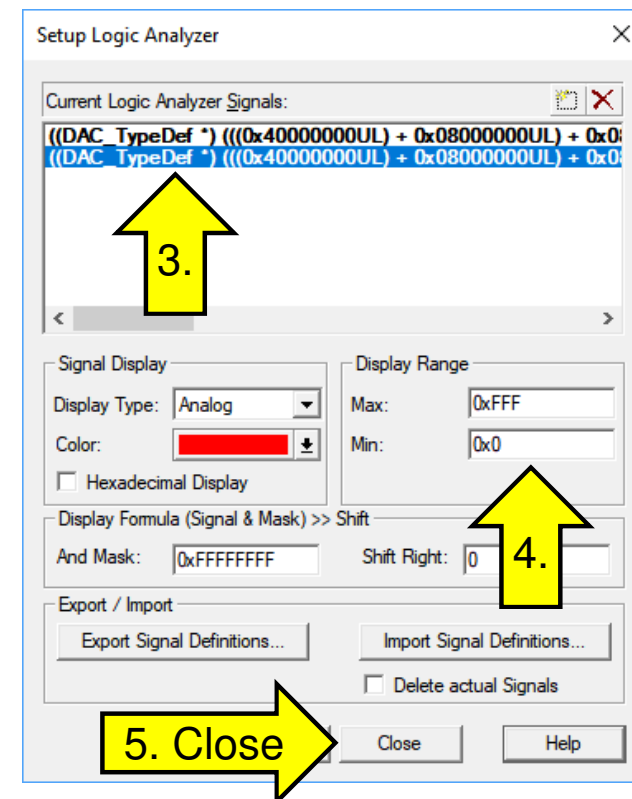


2.

3.

4.

5. Close

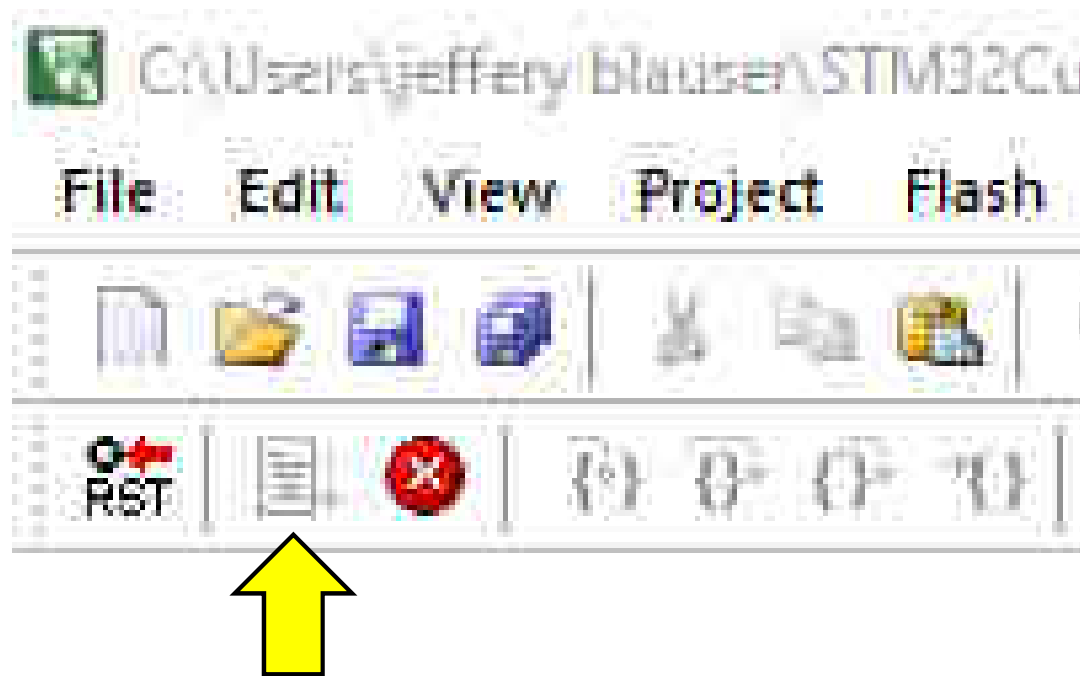




Keil MDK-ARM

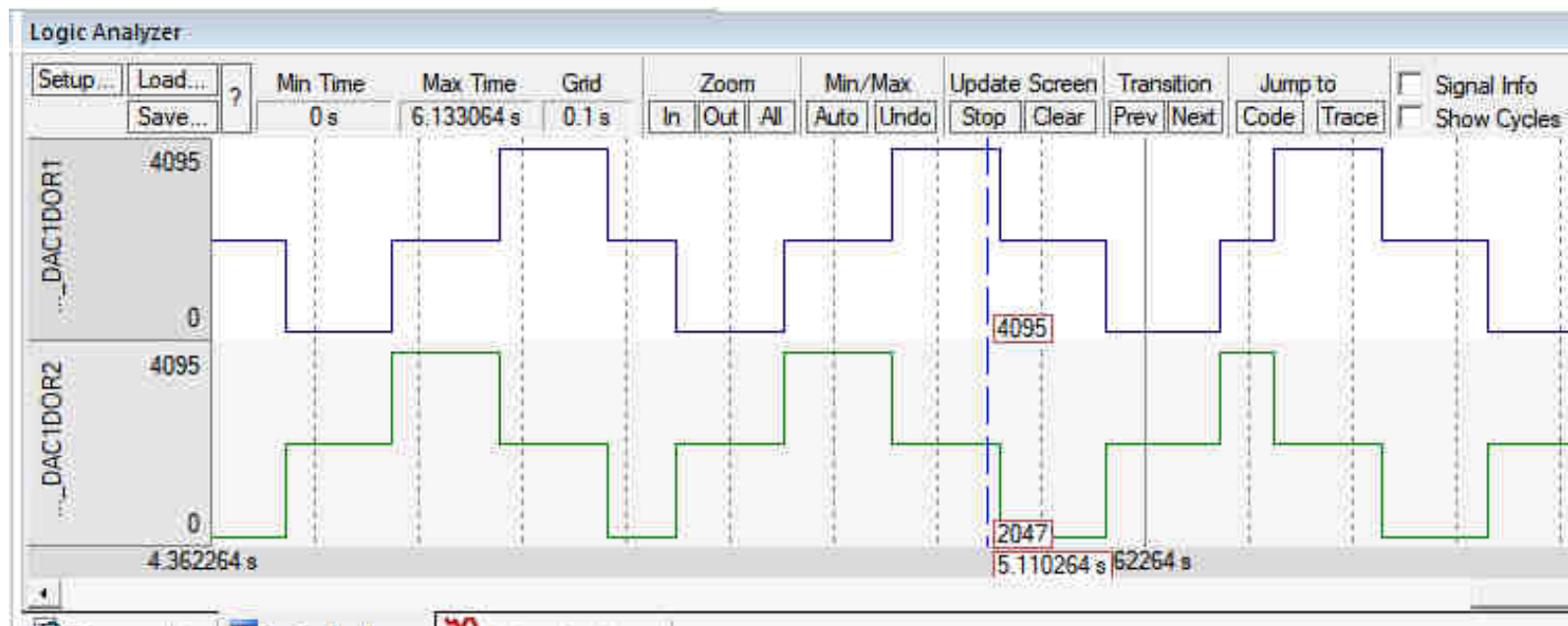
135

RUN



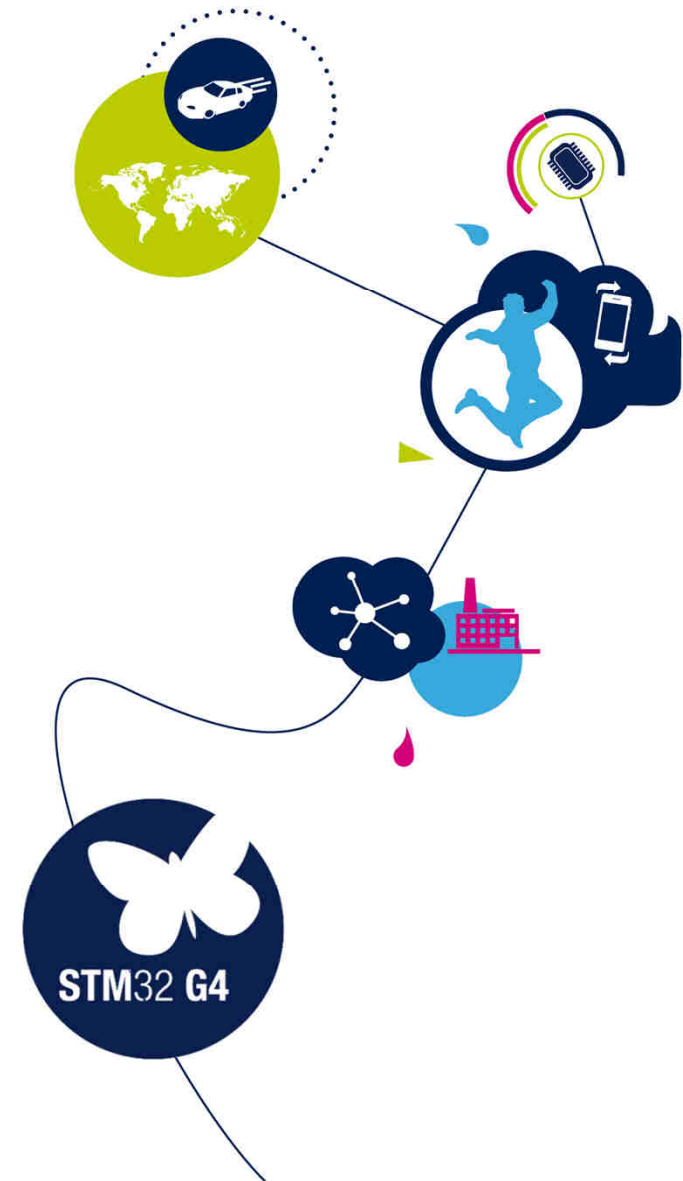


Logic Analyzer Output



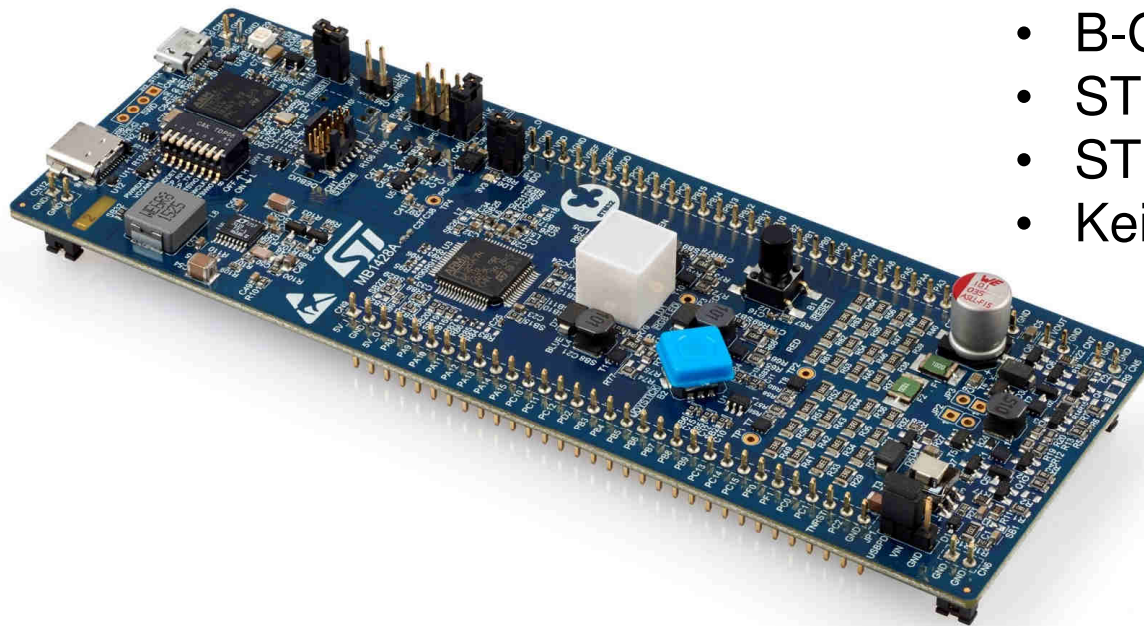
STM32G4 mixed-signal MCU workshop

*Hands On:
Programmable Gain Amplifier*





Systems Check



- USB cable
- B-G474E-DPOW1 Discovery Board
- STM32G4 Family Cube Library
- STM32CubeMx
- Keil MDK-ARM



Programmable Gain Amplifiers

139

G4 PGA

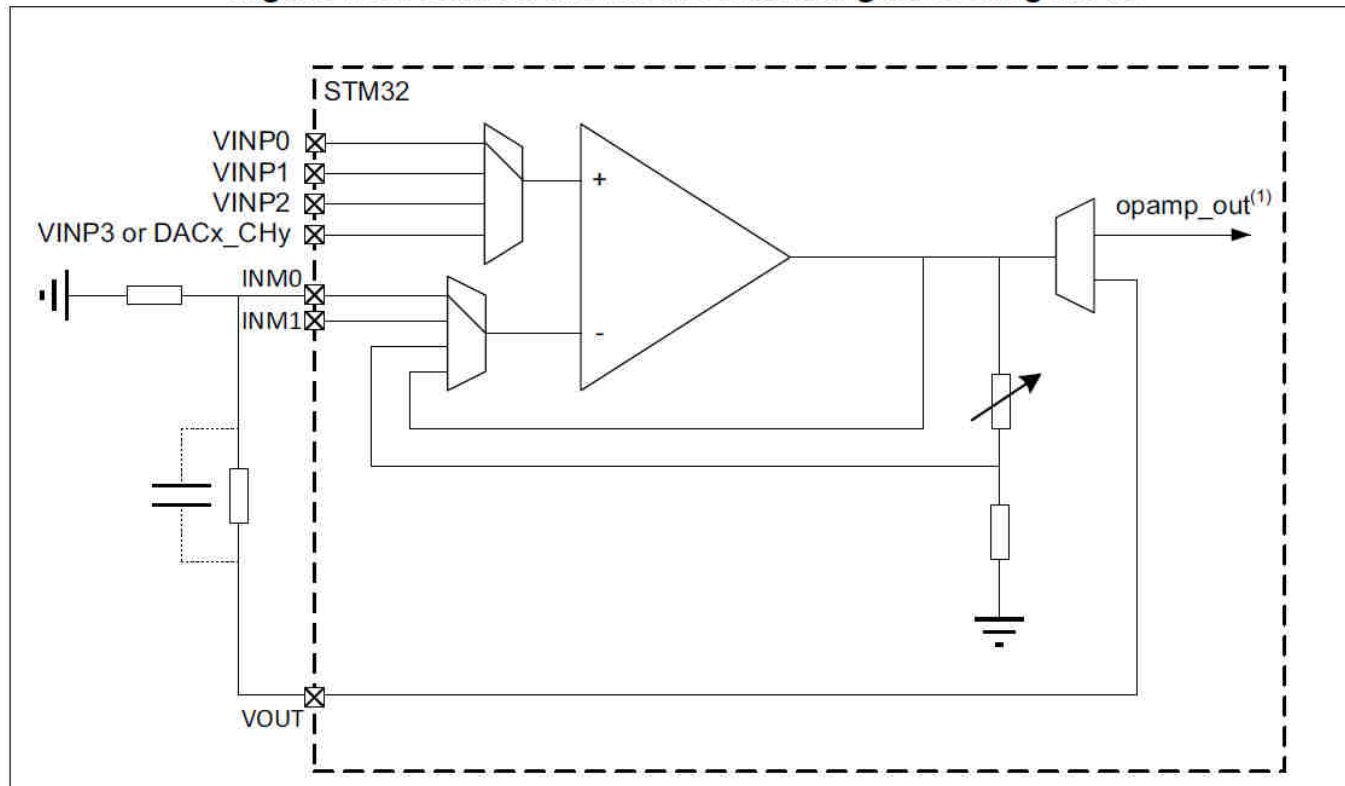
- 6 PGA's available
- Non inverting gain 2/4/8/16/32/64
- Inverting gain -1/-3/-7/-15/-31/-63
- Rail-to-Rail input and output
- Low input bias current
- Low input offset voltage
- 15MHz GBW product
- High speed mode for better slew rate

Internally connected to:

- ADC inputs
- External pins, both inputs and outputs
- Internal resistors
- DAC3 & DAC4 outputs

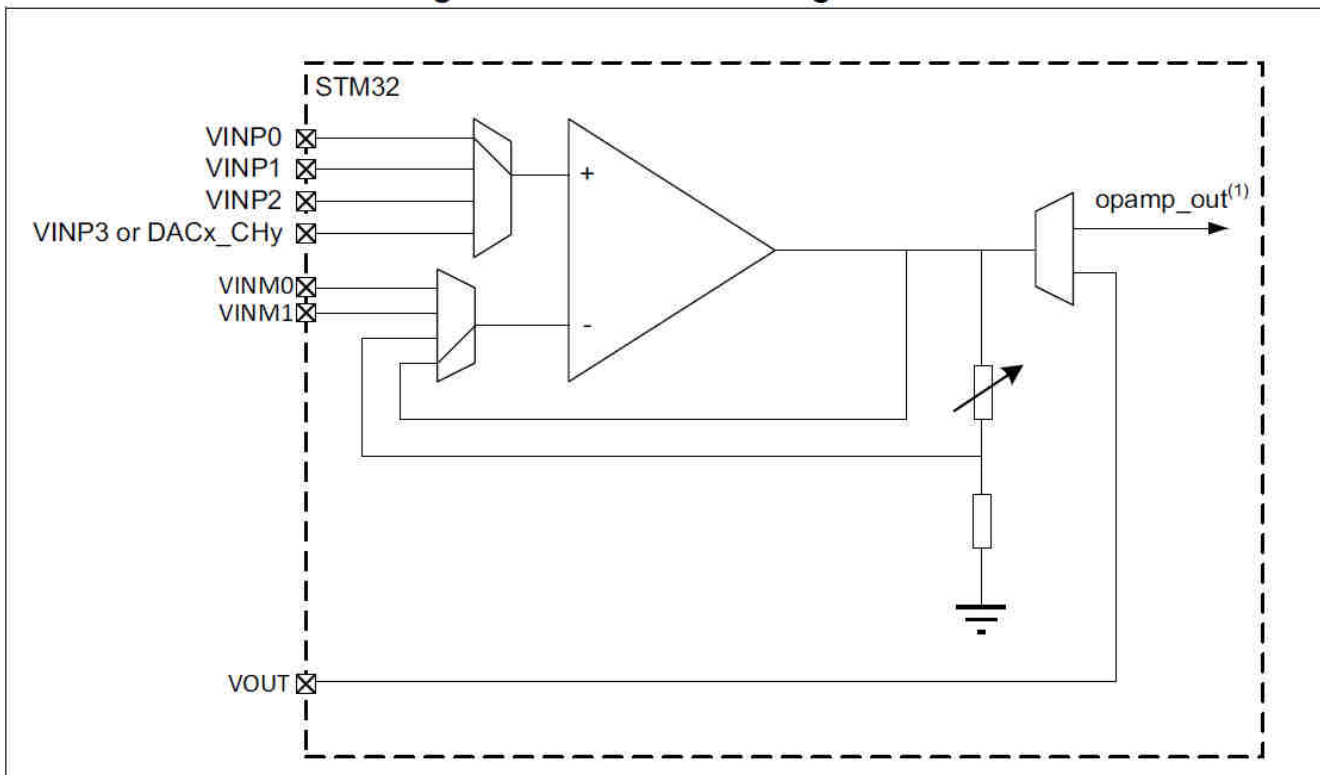
G4 Op Amp Stand Alone Mode

Figure 168. Standalone mode: external gain setting mode



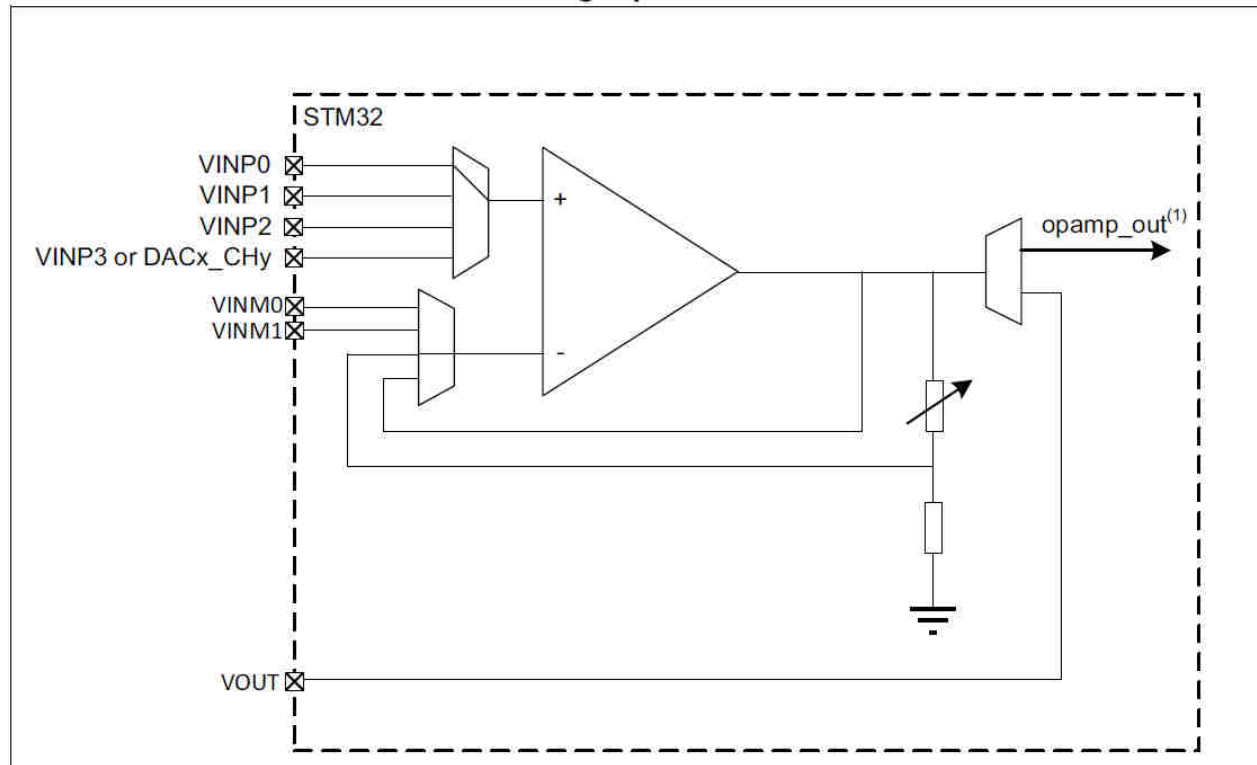
G4 Op Amp Follower Mode

Figure 169. Follower configuration



G4 Op Amp PGA Mode

Figure 170. PGA mode, internal gain setting (x2/x4/x8/x16/x32/x64),
inverting input not used





Lab steps 1 of 2

- The DMA provides samples (sinewave) to the DAC.
 - The DAC peripheral generates a sine wave signal on DAC1_OUT1(PA4) which is amplified by the OPAMP3.
 - The OPAMP3 amplified output is on PB1 with gain of either 2 or 4. The OPAMP gain is changed on the fly while the OPAMP remains enabled.
- The test steps are:
 - Step 1:
 - DAC: normal power mode
 - OPAMP: normal power mode with gain = 2
 - DMA: circular mode - DMA half transfer IT handled by CPU



Lab steps 2 of 2

- Step 2:

DAC: normal power mode

OPAMP: normal power mode

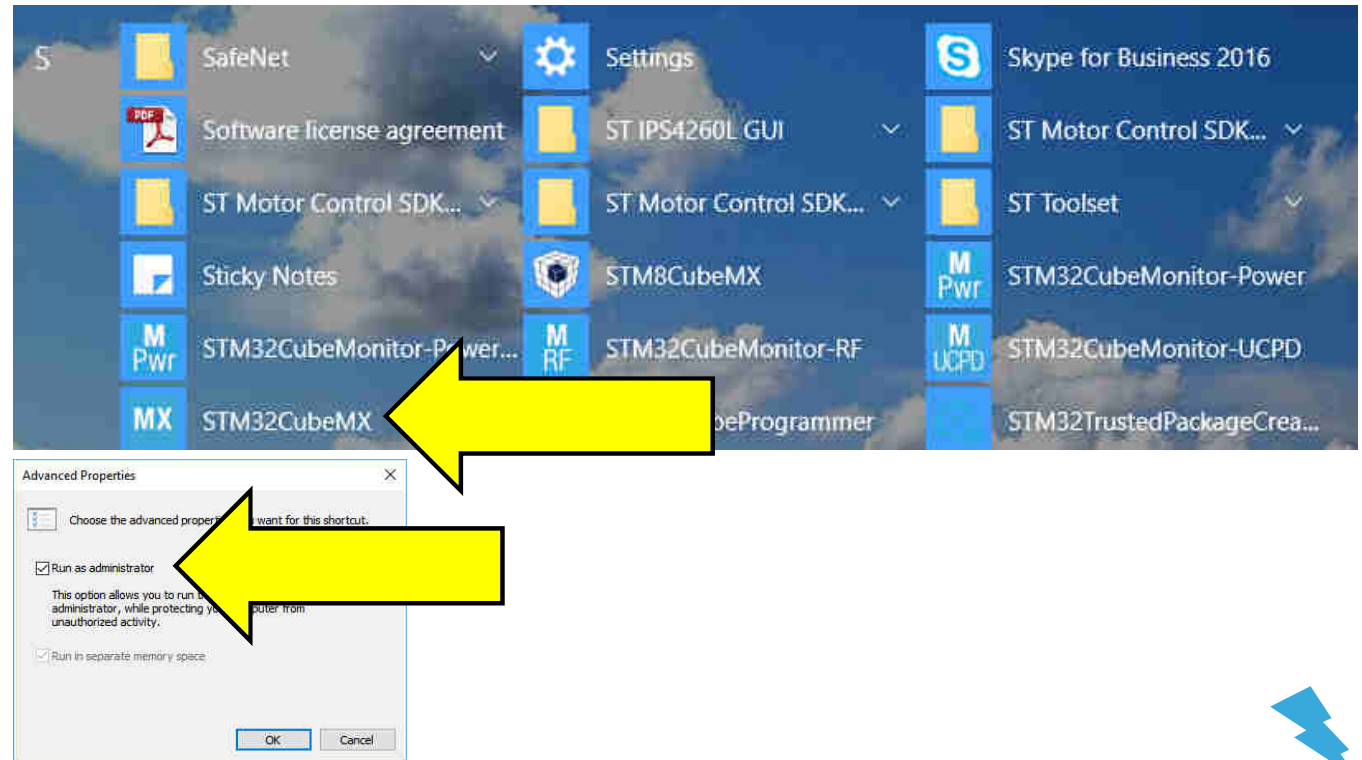
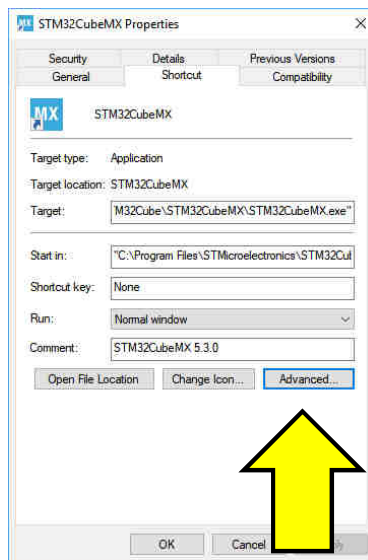
- gain is changed from 2 to 4 or from 4 to 2

DMA: circular mode - DMA half transfer IT handled by CPU

We will use the built in Logic Analyzer today instead of an o'scope on the DACs



Start up





Splash Page – G4 Library Installed?

The screenshot displays the STM32CubeMX software interface. On the left, the 'Embedded Software Packages Manager' window is open, showing a table of installed and available software packages. A yellow arrow points to the 'STM32G4' section of the table. The main window shows the 'Manage software installations' section with a 'CHECK FOR UPDATES' button and an 'INSTALL / REMOVE' button. A large yellow arrow points to the 'INSTALL / REMOVE' button with the text 'Install/Remove'.

Description	Installation	Available Version
STM32F7		
STM32G0		
STM32G4		
STM32Cube MCU Package for STM32G4 Series	1.1.0	1.1.0
STM32Cube MCU Package for STM32G4 Series	1.0.1	1.0.1

Details

From Local From Url Refresh Install Now Remove Now Close

Manage software installations

Check for STM32CubeMX and embedded software packages updates

CHECK FOR UPDATES

Install or remove embedded software packages

INSTALL / REMOVE

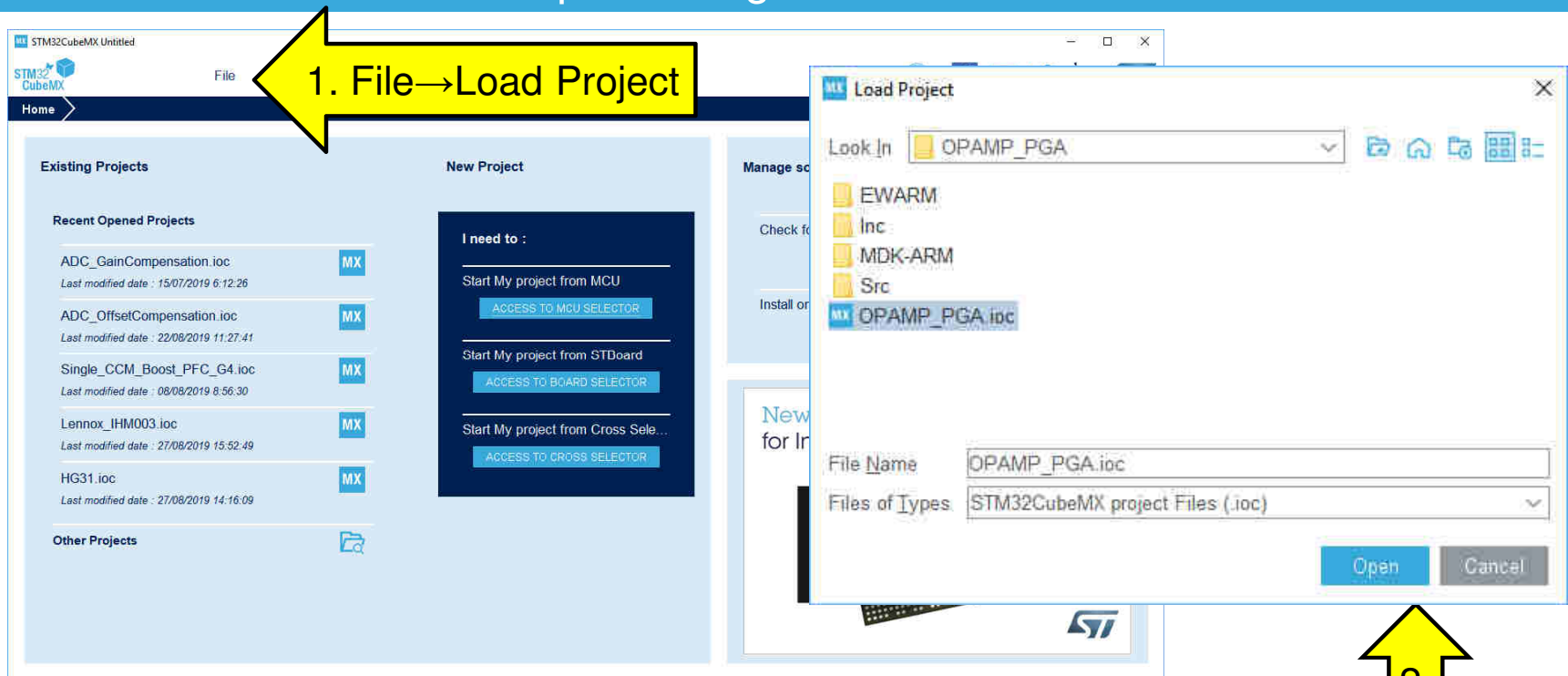
New multicore STM32MP1 Series for Industrial and IoT applications

STM32MP1

OpenSTLinux Distribution



Splash Page



C:\...\STM32Cube\Repository\STM32Cube_FW_G4_V1.1.0\Projects\NUCLEO-G474RE\Examples\OPAMP\OPAMP_PGA

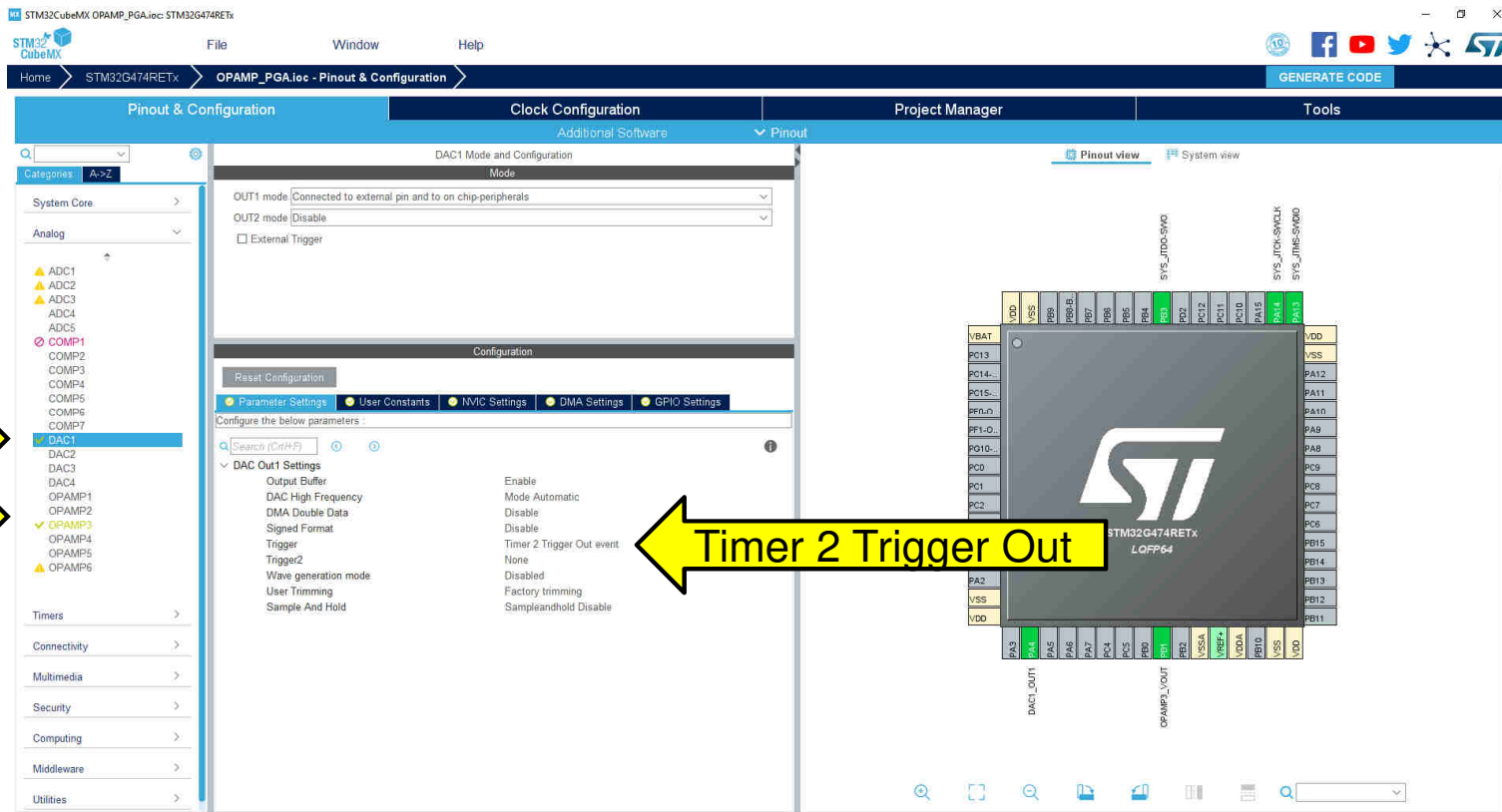


Graphic Design Representation – Click on Analog>





DAC1, Timer 2, Op Amp 3





Project Manager – Keil MDK-ARM V5

STM32CubeMX OPAMP_PGA.ioc: STM32G474RETx

File Window Help

Home > STM32G474RETx > OPAMP_PGA.ioc - Project Manager > GENERATE CODE

Pinout & Configuration Clock Configuration Project Manager Tools

Project Settings

Project Name
OPAMP_PGA

Project Location
ery blausen\STM32Cube\Repository\STM32Cube_FW_G4_V1.1.0\Projects\B-G474E-DPOW1\Examples\OPAMP

Application Structure
Basic ☐ Do not generate the main()

Toolchain Folder Location
I32Cube\Repository\STM32Cube_FW_G4_V1.1.0\Projects\B-G474E-DPOW1\Examples\OPAMP\OPAMP_PGA\

Toolchain / IDE
MDK-ARM V5

Linker Settings

Minimum Heap Size 0x200

Minimum Stack Size 0x400

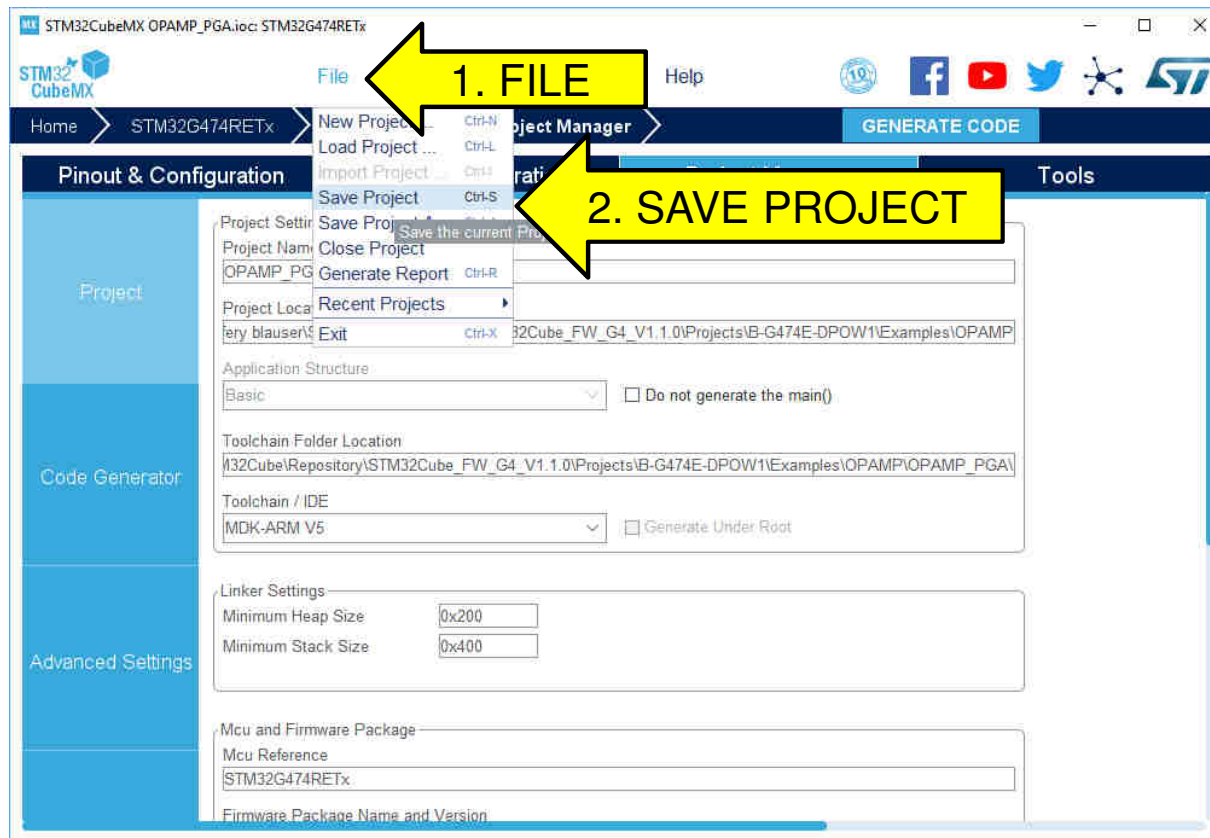
Mcu and Firmware Package

Mcu Reference
STM32G474RETx

Firmware Package Name and Version



FILE→SAVE PROJECT





Generate Code→Open Project

The screenshot illustrates the process of generating code and opening a project in STM32CubeMX. The main window shows the 'Project Manager' tab with the 'GENERATE CODE' button highlighted by a yellow arrow labeled '1.'. Below this, the 'Code Generation' dialog box is open, displaying a message: 'The Code is successfully generated under C:/Users/jeffery blausen/STM32Cube/Repository/STM32Cube_FW_G4_V1.1.0/Projects/B-G474E-DPOW1/Examples/OPAMP/OPAMP_PGA'. The dialog box includes buttons for 'Open Folder', 'Open Project', and 'Close'. A yellow arrow labeled '2.' points to the 'Open Project' button. The background window also shows the 'Advanced Settings' tab with fields for 'MDK-ARM V5', 'Linker Settings' (Minimum Heap Size: 0x200, Minimum Stack Size: 0x400), and 'Mcu and Firmware Package' (Mcu Reference: STM32G474RETx).



Keil Start up page

```
143 if (UserButtonStatus == 1)
144 {
145     HAL_Delay(200);
146     UserButtonStatus = 0;
147     /* Change the gain */
148     if (hopamp3.Init.PgaGain == OPAMP_PGA_GAIN_2_OR_MINUS_1)
149     {
150         hopamp3.Init.PgaGain = OPAMP_PGA_GAIN_4_OR_MINUS_3;
151     }
152     else
153     {
154         hopamp3.Init.PgaGain = OPAMP_PGA_GAIN_2_OR_MINUS_1;
155     }
156     /* Update OPAMP config */
157     /* Gain is changed on the fly */
158     if (HAL_OK != HAL_OPAMP_Init(&hopamp3))
159     {
160         Error_Handler();
161     }
162 }
163 }
164 /* USER CODE END WHILE */
165
166 /* USER CODE BEGIN 3 */
167 /* USER CODE END 3 */
168 }
169
170 /**
171  * @brief System Clock Configuration
172  * @retval None
173  */
174 void SystemClock_Config(void)
175 {
176     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
177     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
178
179     /* Configure the main internal regulator output voltage
180     */
181     HAL_PWREX_ControlVoltageScaling(FWR_REGULATOR_VOLTAGE_SCALE1_BOOST);
182     /* Initializes the CPU, AHB and APB buses clocks
183     */
184     RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
185     RCC_OscInitStruct.HSISource = RCC_HSI_ON;
186     RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
187     RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
```

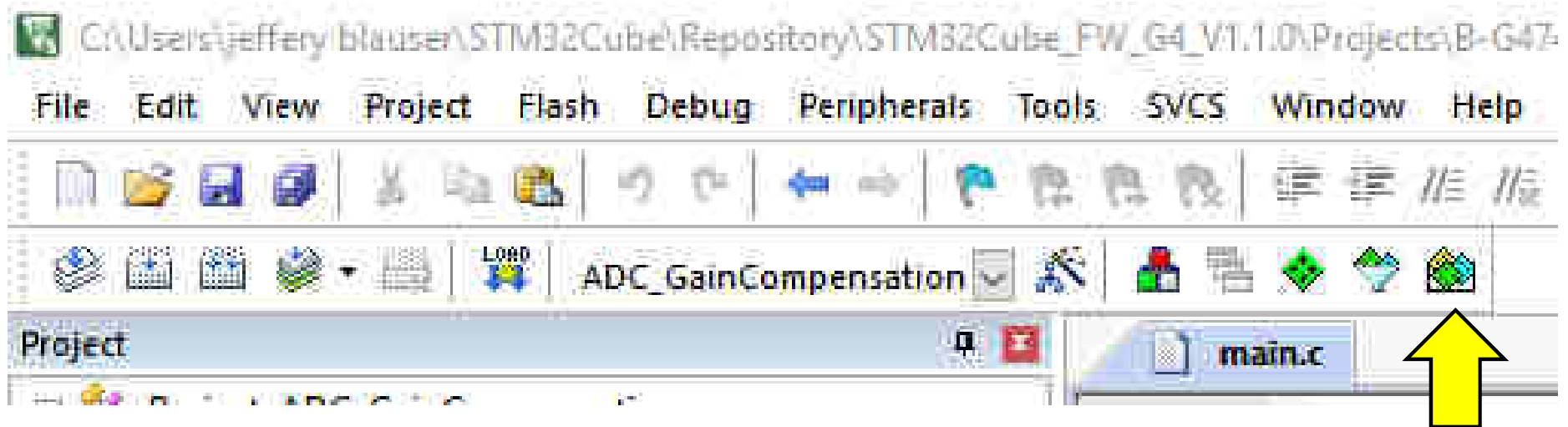




Keil MDK-ARM

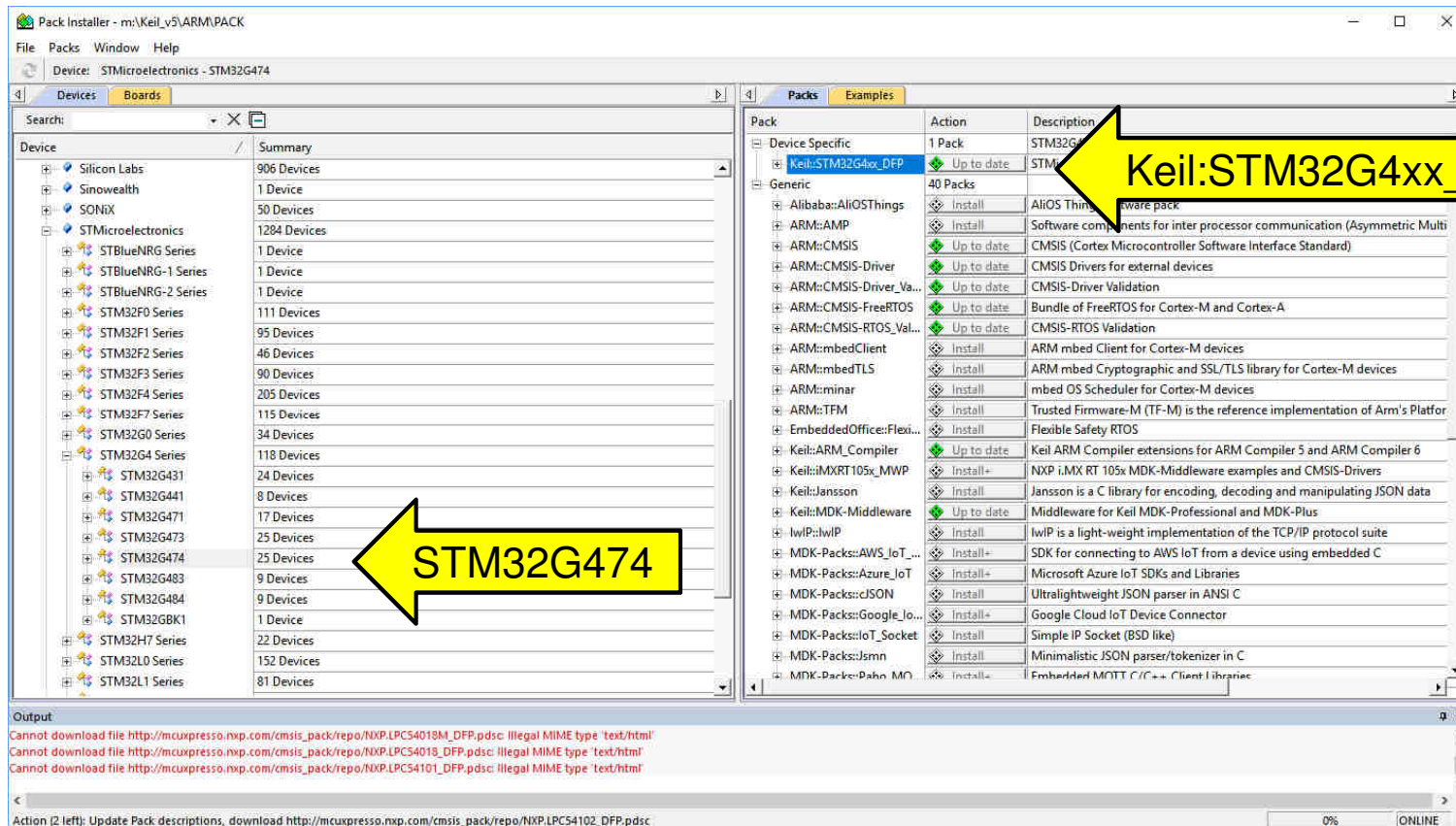
154

Check the Device Pack





G4 Device Pack

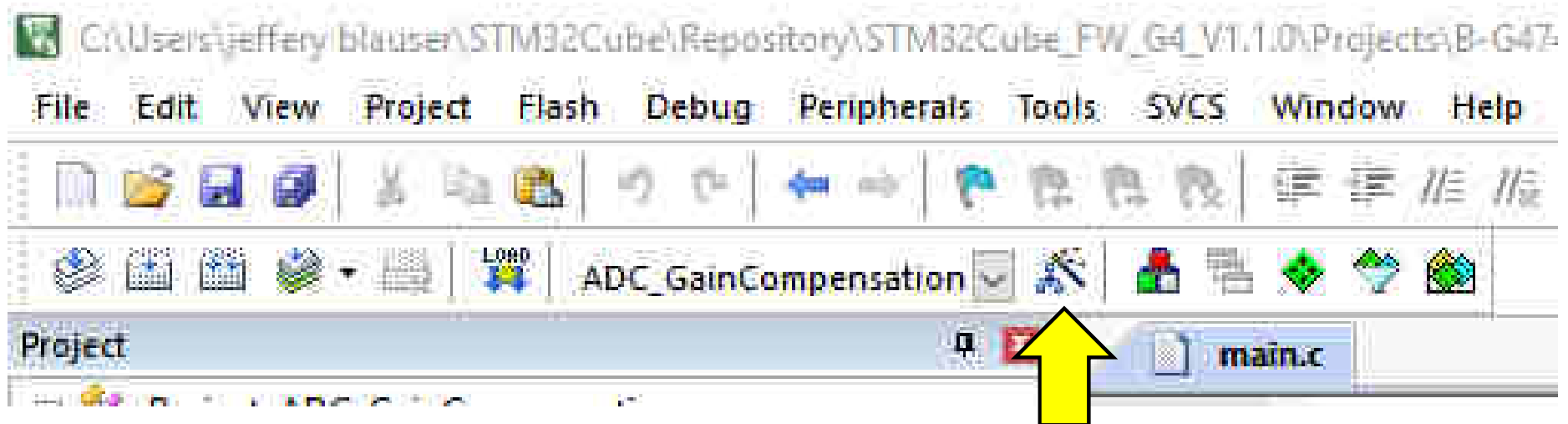




Keil MDK-ARM

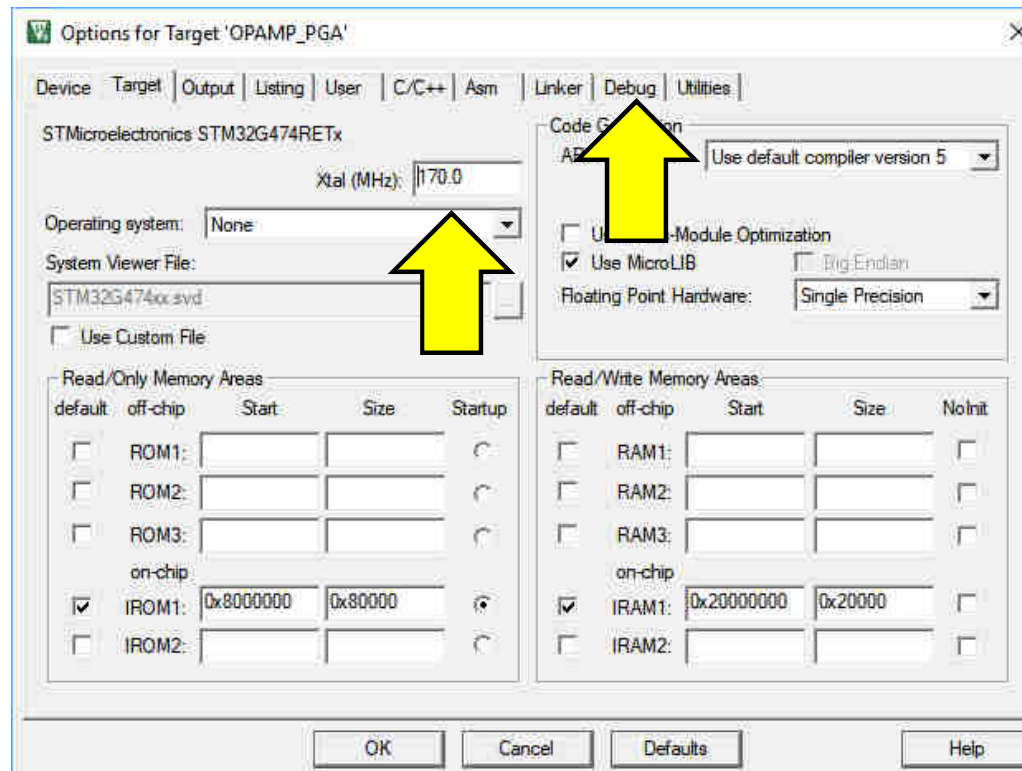
156

Check the Options



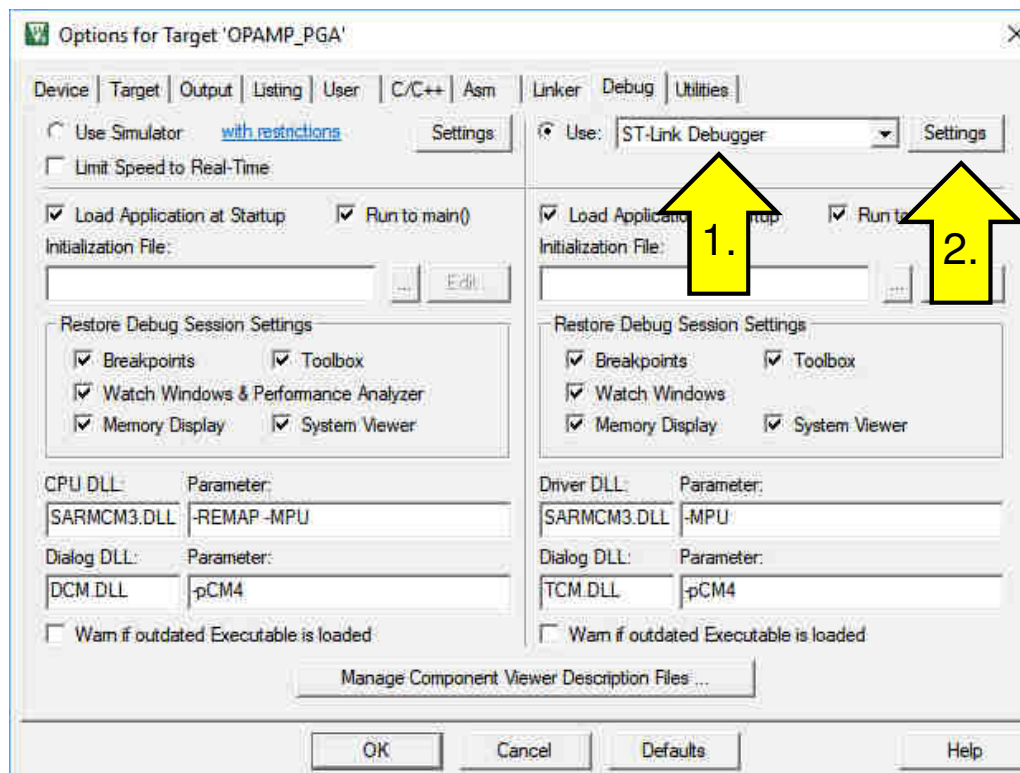


Clock speed and Debug



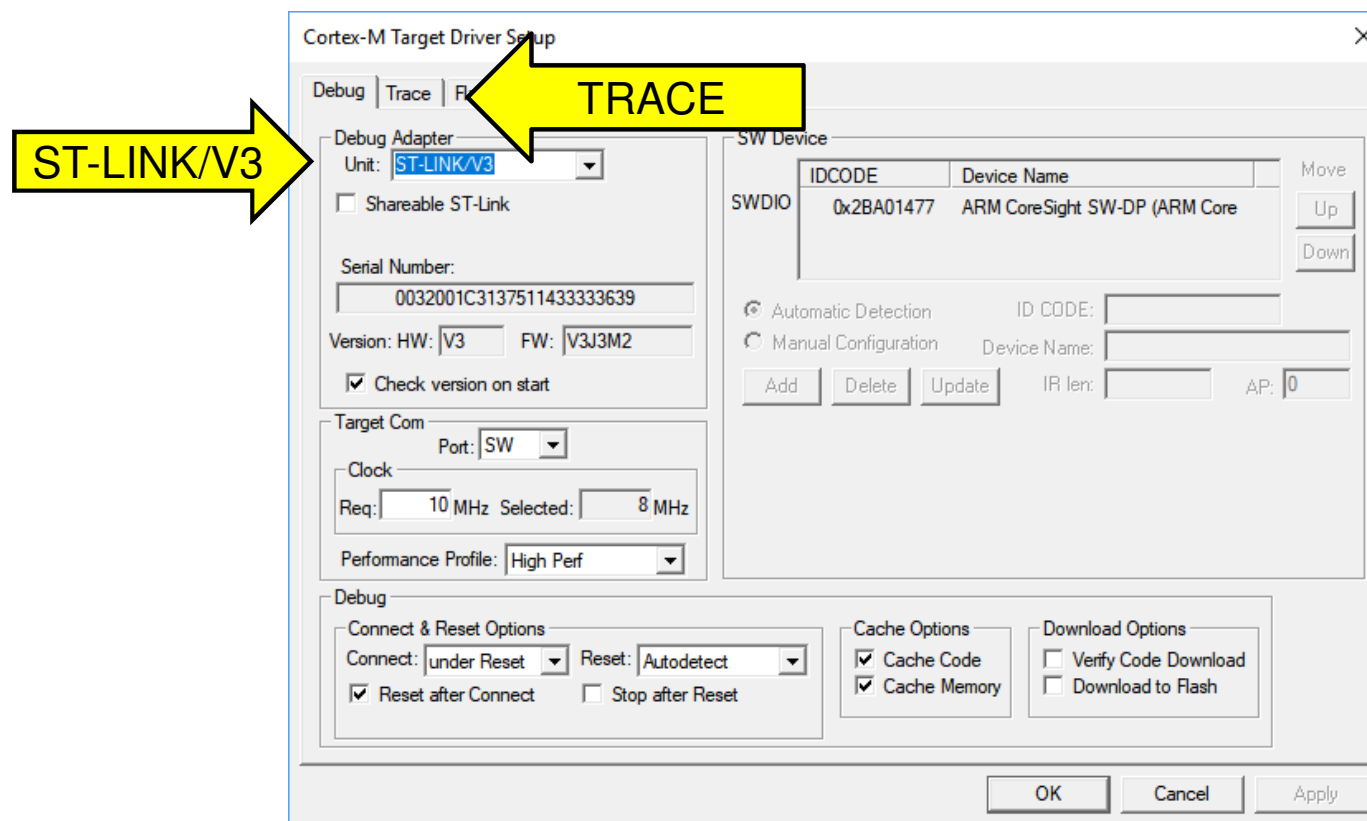


ST-Link Debugger - Settings





ST-Link/V3 & TRACE





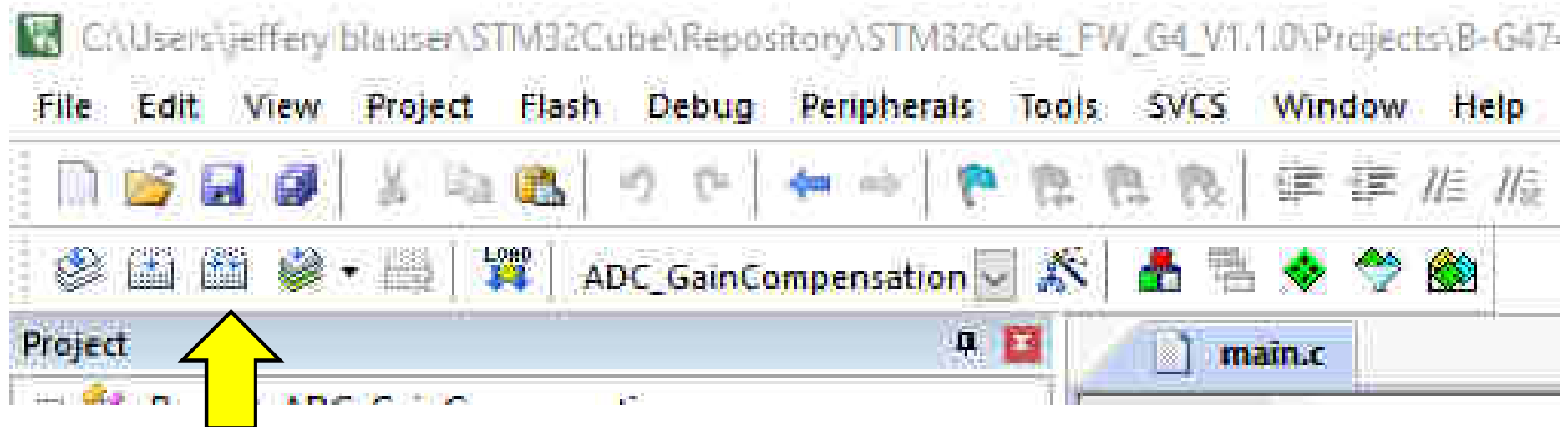
Core Clock & TRACE Enable (After)

The screenshot shows the 'Cortex-M Target Driver Setup' dialog box with the 'Trace' tab selected. The following settings are visible:

- Core Clock:** 170.000000 MHz (Annotated with '1. 170')
- Trace Enable:** ☒ (Annotated with '2. TRACE Enable')
- Trace Port:** Serial Wire Output - UART/NRZ
- SWO Clock Prescaler:** 15
- Autodetect:** ☒
- SWO Clock:** 11.333333 MHz
- Timestamps:** ☒ Enable, Prescaler: 1
- PC Sampling:** Prescaler: 1024*16, ☐ Periodic, Period: <Disabled>, ☐ on Data R/W Sample
- Trace Events:** ☐ CPI: Cycles per Instruction, ☐ EXC: Exception overhead, ☐ SLEEP: Sleep Cycles, ☐ LSU: Load Store Unit Cycles, ☐ FOLD: Folded Instructions, ☐ EXCTRC: Exception Tracing (Annotated with '3. Uncheck EXCTRC')
- ITM Stimulus Ports:** Enable: 0x00008000, Privilege: 0x00000008. The 'Port 15' checkbox is checked, while others are unchecked (Annotated with '4. Uncheck all but 15').
- Advanced settings:** ☐ Ignore packets with no SYNC, ☐ Overwrite CYCCNT
- Buttons:** OK, Cancel, Apply (Annotated with '5. OK' pointing to the OK button)



Build

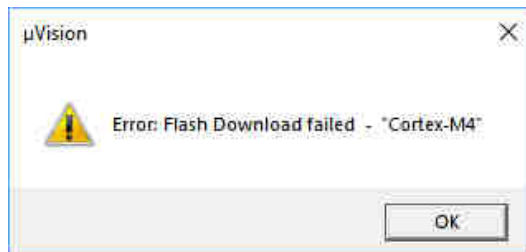
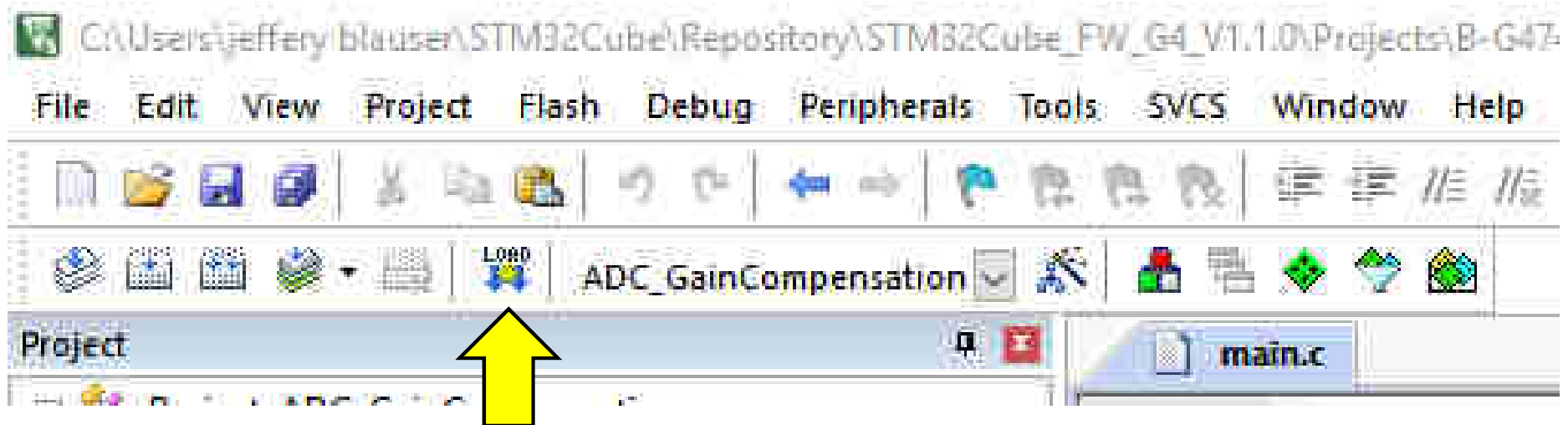


Build Output

```
Program Size: Code=6516 RO-data=596 RW-data=44 ZI-data=1268
FromELF: creating hex file...
OPAMP_PGA\OPAMP_PGA.hex: Warning: Q9931W: Your license for feature Keil will expire in 30 days
Finished: 0 information, 1 warning and 0 error messages.
"OPAMP_PGA\OPAMP_PGA.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:01:22
```



Load



Build Output

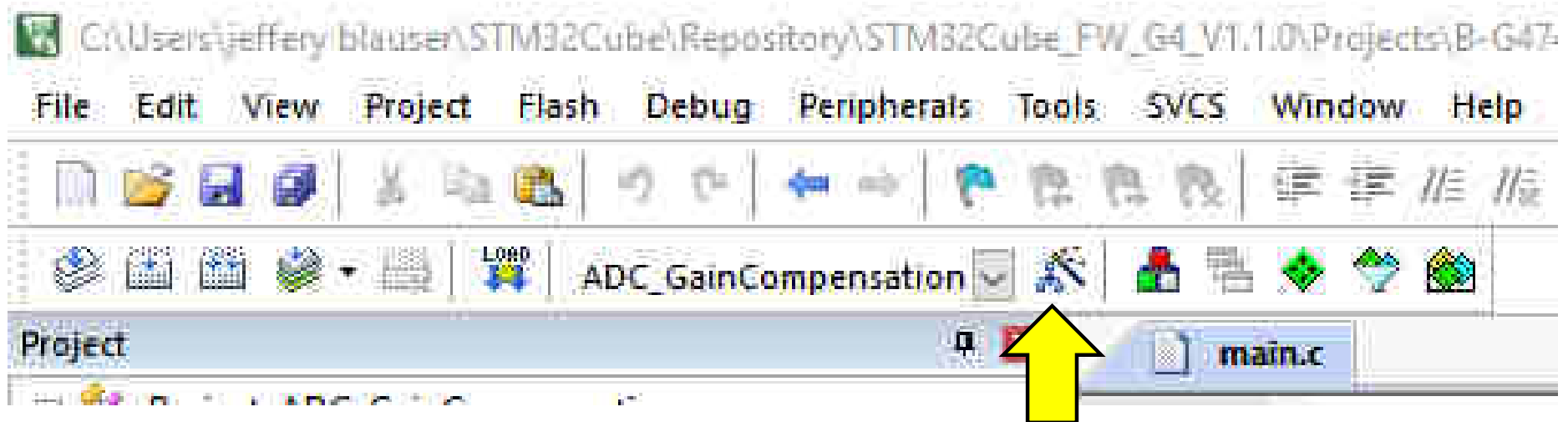
```
Contents mismatch at: 08001860H (Flash=38H Required=C5H) !
Contents mismatch at: 08001861H (Flash=46H Required=F8H) !
Contents mismatch at: 08001862H (Flash=FEH Required=88H) !
Contents mismatch at: 08001863H (Flash=F7H Required=00H) !
Contents mismatch at: 08001864H (Flash=B9H Required=60H) !
Contents mismatch at: 08001865H (Flash=FCH Required=6CH) !
Too many errors to display !
Error: Flash Download failed - "Cortex-M4"
```



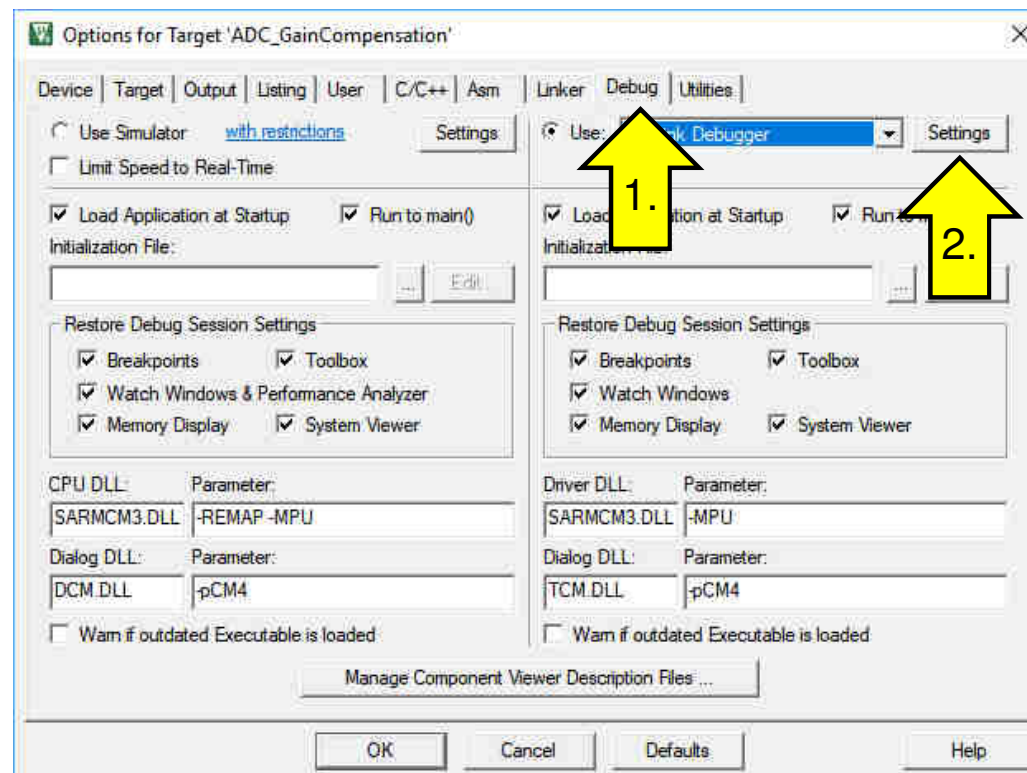
Keil MDK-ARM

163

Options

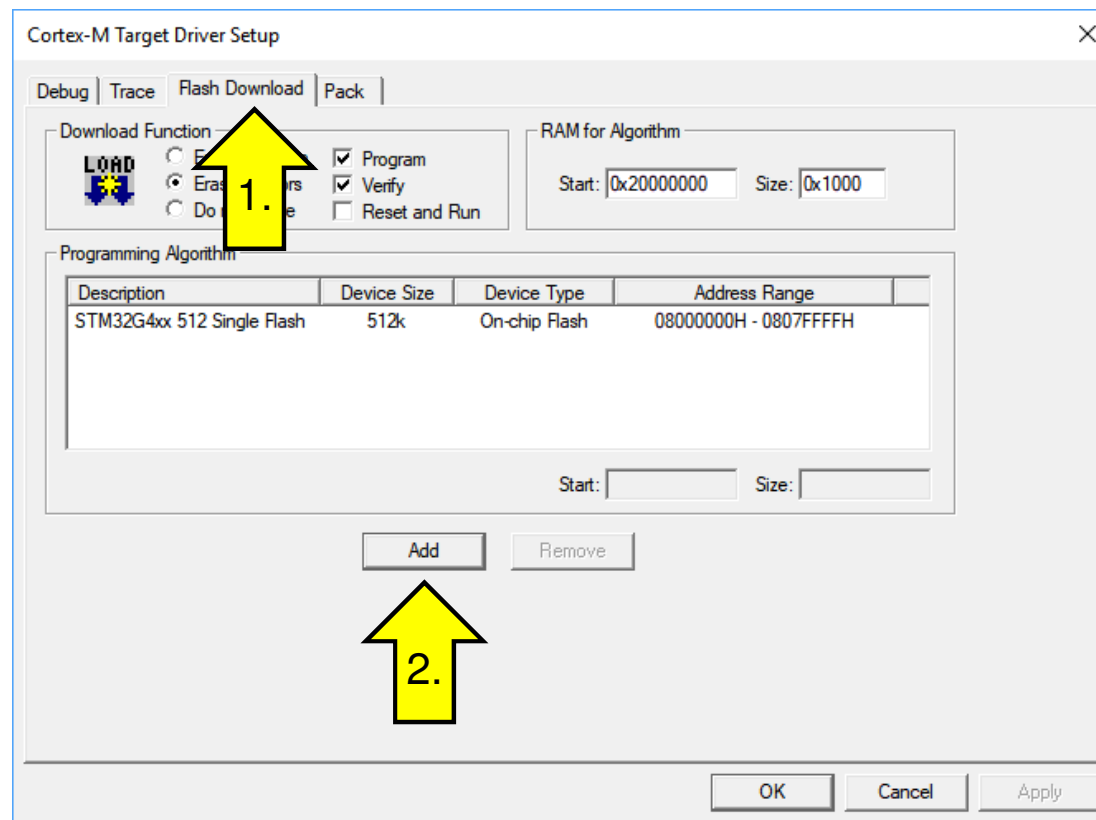


ST-Link Debugger - Settings



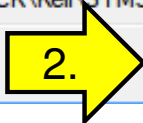
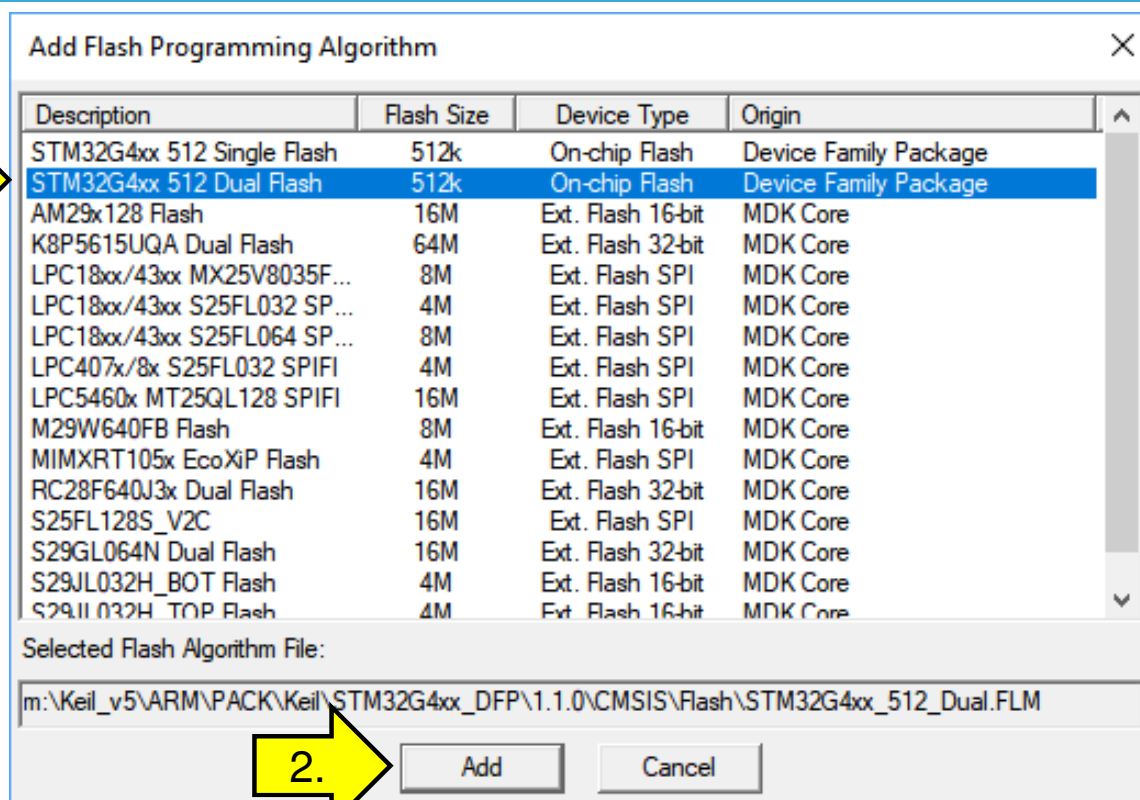


Flash Download → Add

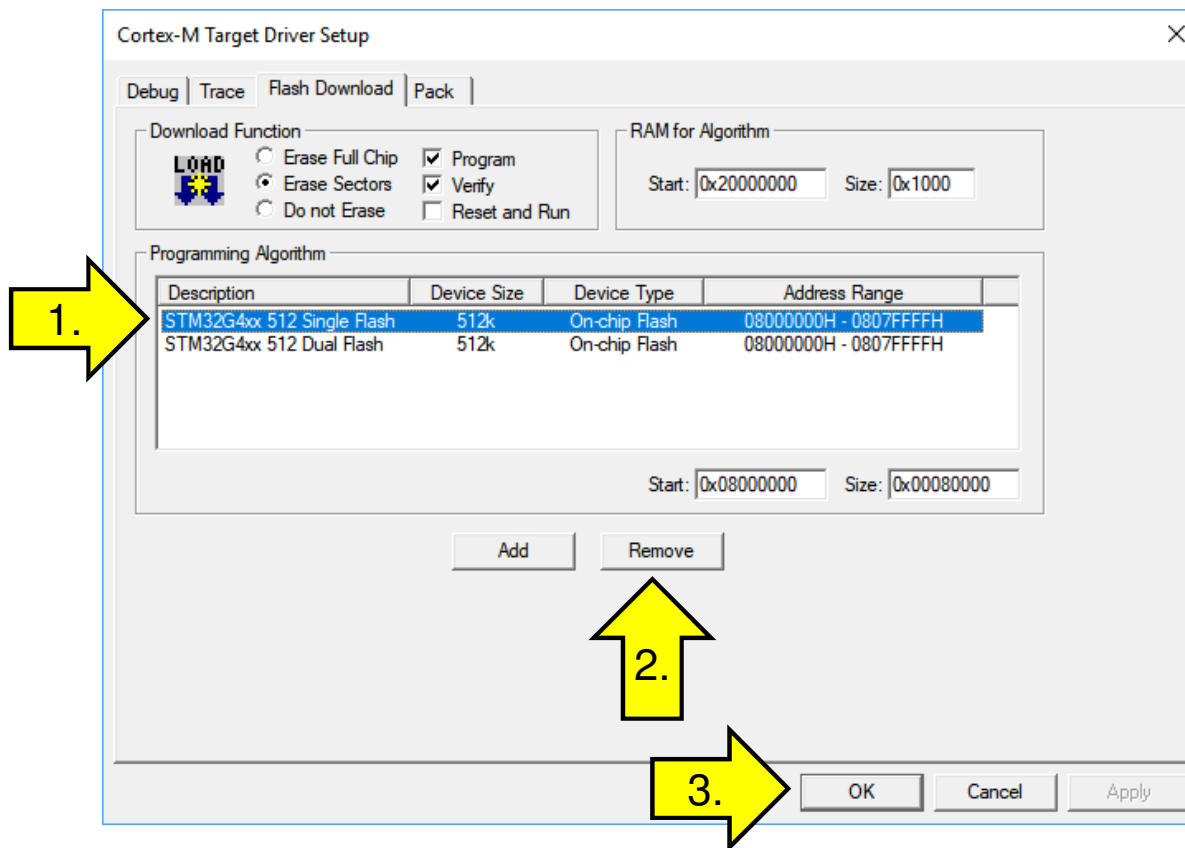




STM32G4xx 512 Dual Flash

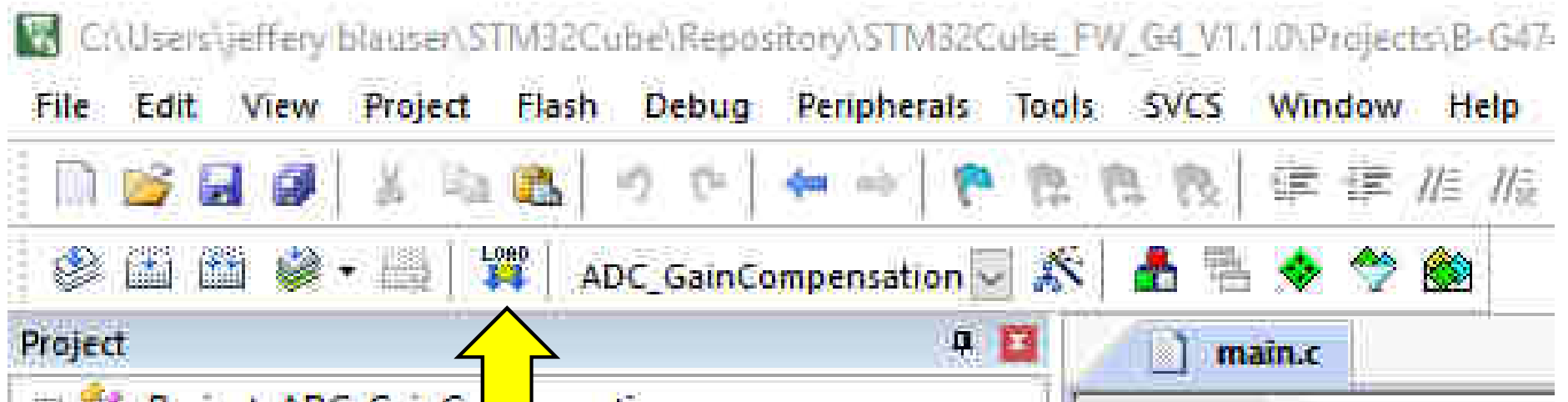


Remove Single Flash





Load



Build Output

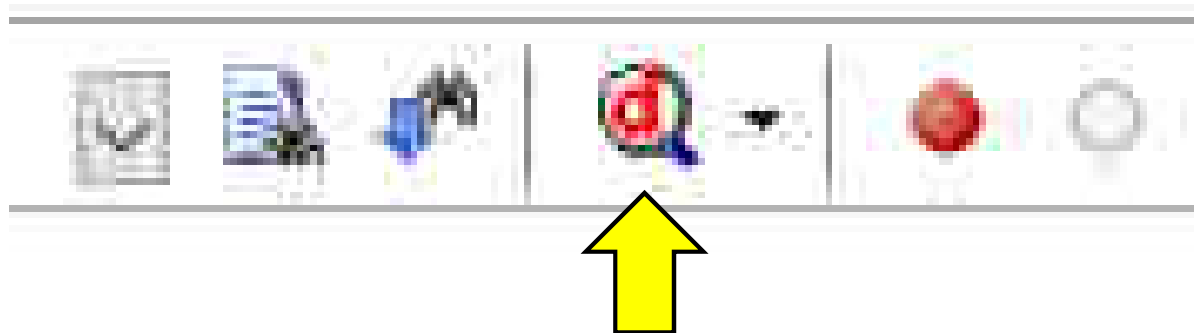
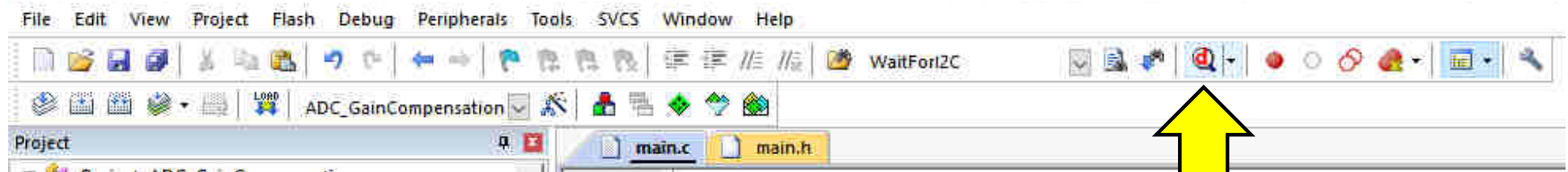
```
Build Time Elapsed: 00:01:22
Load "OPAMP_PGA\\OPAMP_PGA.axf"
Erase Done.
Programming Done.
Verify OK.
Flash Load finished at 17:34:57
```




Keil MDK-ARM

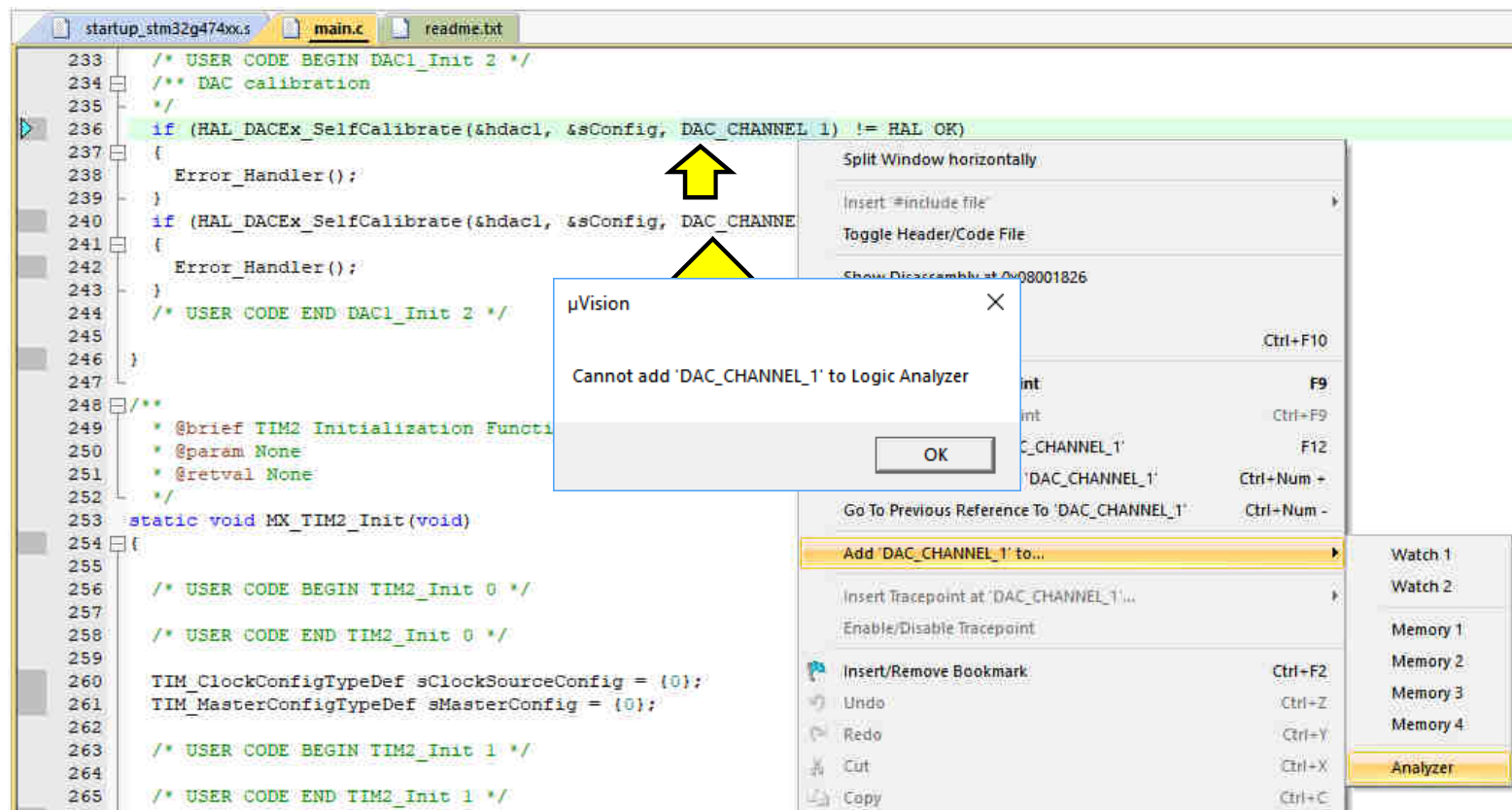
169

Debugger





Place variable in the Analyzer – main.c about lines 235 & 240

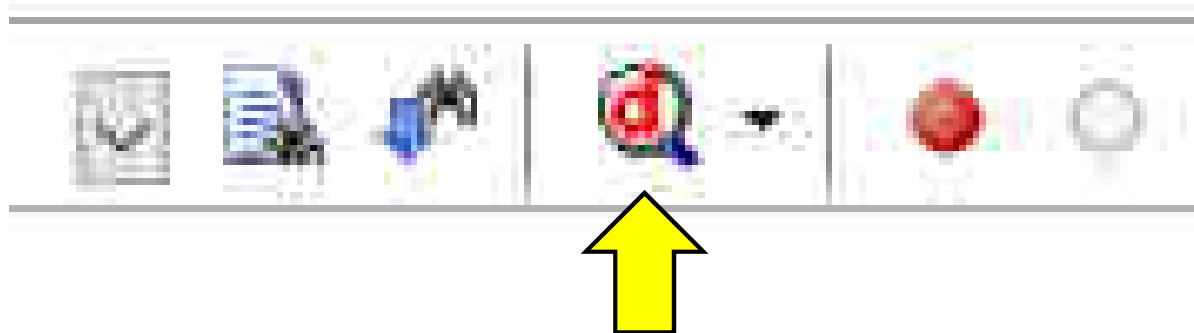
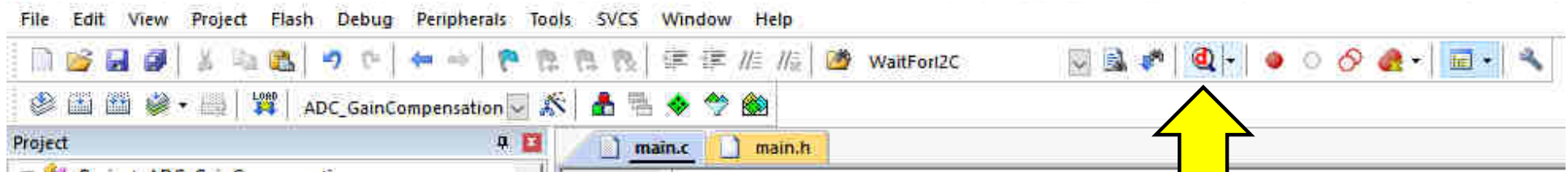




Keil MDK-ARM

171

Exit Debugger





Add 3 new variables in main.c about line 59

```
54  /* USER CODE BEGIN PV */
55  /* Private variables -----
56  /* set to 1 after User push-button interrupt */
57  __IO uint32_t UserButtonStatus = 0;
58
59  uint16_t  Display_DAC1DOR1;
60  uint16_t  Display_DAC3DOR2;
61  uint16_t  Display_ADC3IN1;
```

See CodeLinesForOpAmp_09032019.txt in
c:\2019_STM32G4_Workshop\HandsOn_Labs\OPAMP



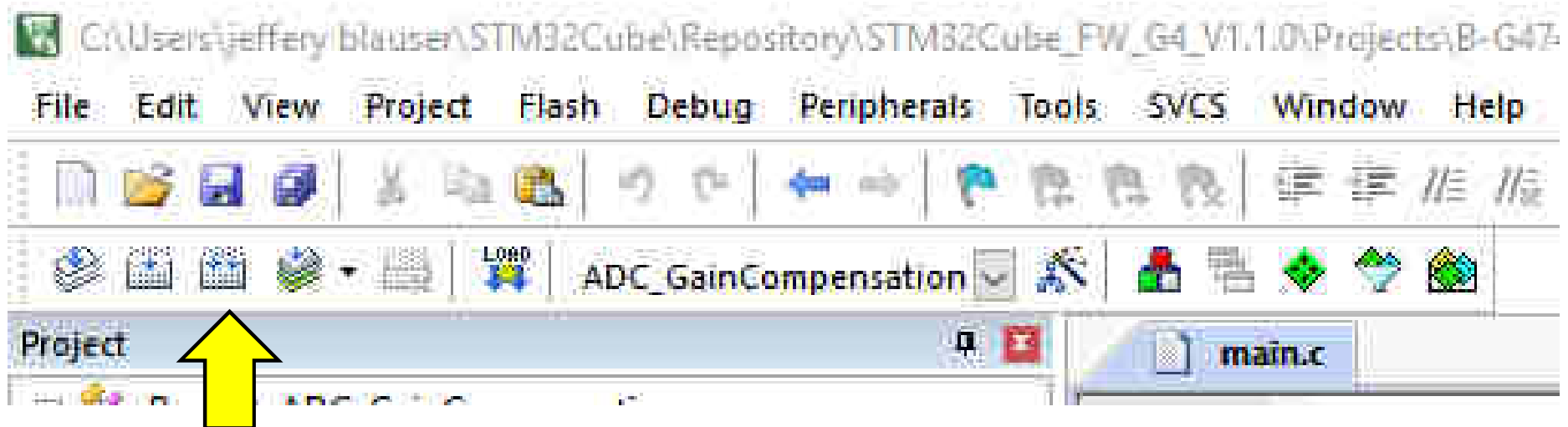
Add new lines in main.c about line 177

```
174     Error_Handler();
175 }
176 }
177 /* Display the global variables in the Logic Analyzer */
178 Display_DAC1DOR1 = DAC1->DOR1;
179 HAL_Delay(1);    //1 millisecond delay to transfer data to the PC
180 Display_DAC3DOR2 = DAC3->DOR1;
181 HAL_Delay(1);    //1 millisecond delay to transfer data to the PC
182 Display_ADC3IN1 = ADC3->DR;
183 HAL_Delay(1);    //1 millisecond delay to transfer data to the PC
184 }
185 /* USER CODE END WHILE */
```

See CodeLinesForOpAmp_09032019.txt in
c:\2019_STM32G4_Workshop\HandsOn_Labs\OPAMP



Build



Build Output

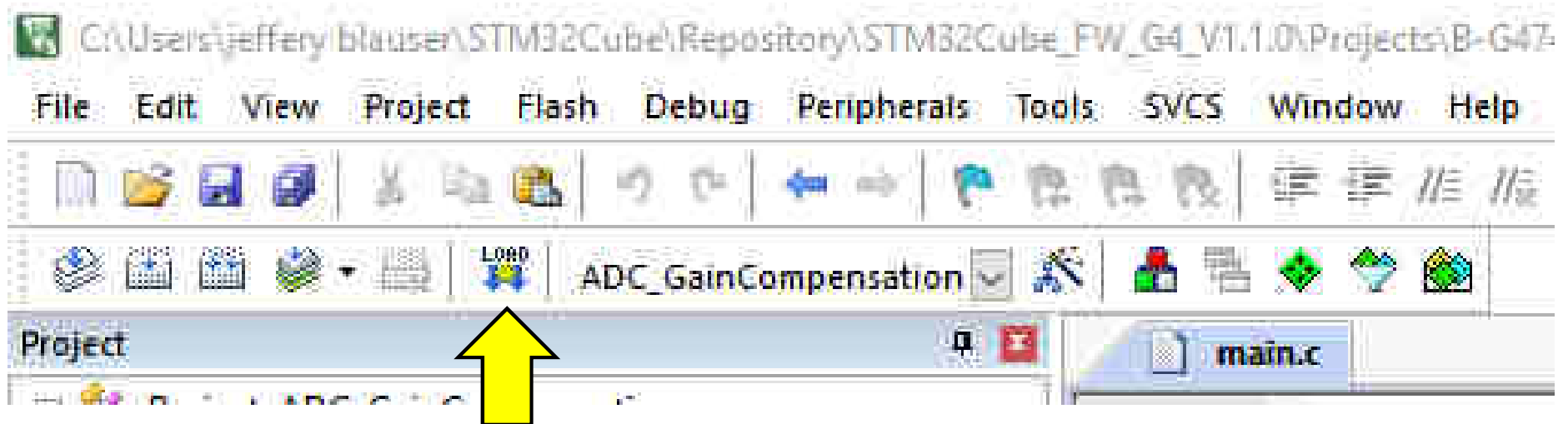
```
Program Size: Code=6516 RO-data=596 RW-data=44 ZI-data=1268
FromELF: creating hex file...
OPAMP_PGA\OPAMP_PGA.hex: Warning: Q9931W: Your license for feature Keil will expire in 30 days
Finished: 0 information, 1 warning and 0 error messages.
"OPAMP_PGA\OPAMP_PGA.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:01:22
```



Keil MDK-ARM

175

Load

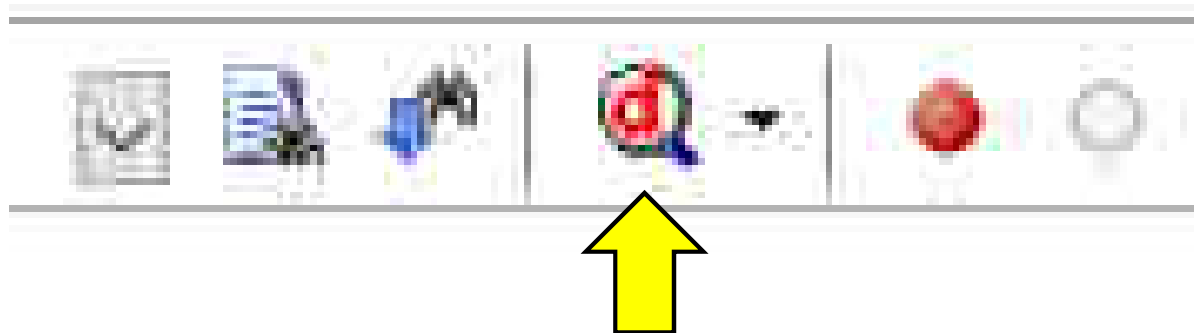
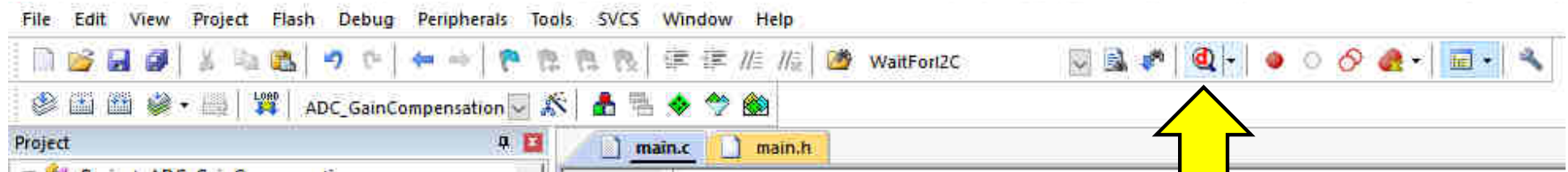




Keil MDK-ARM

176

Debugger – Exit back to the Editor



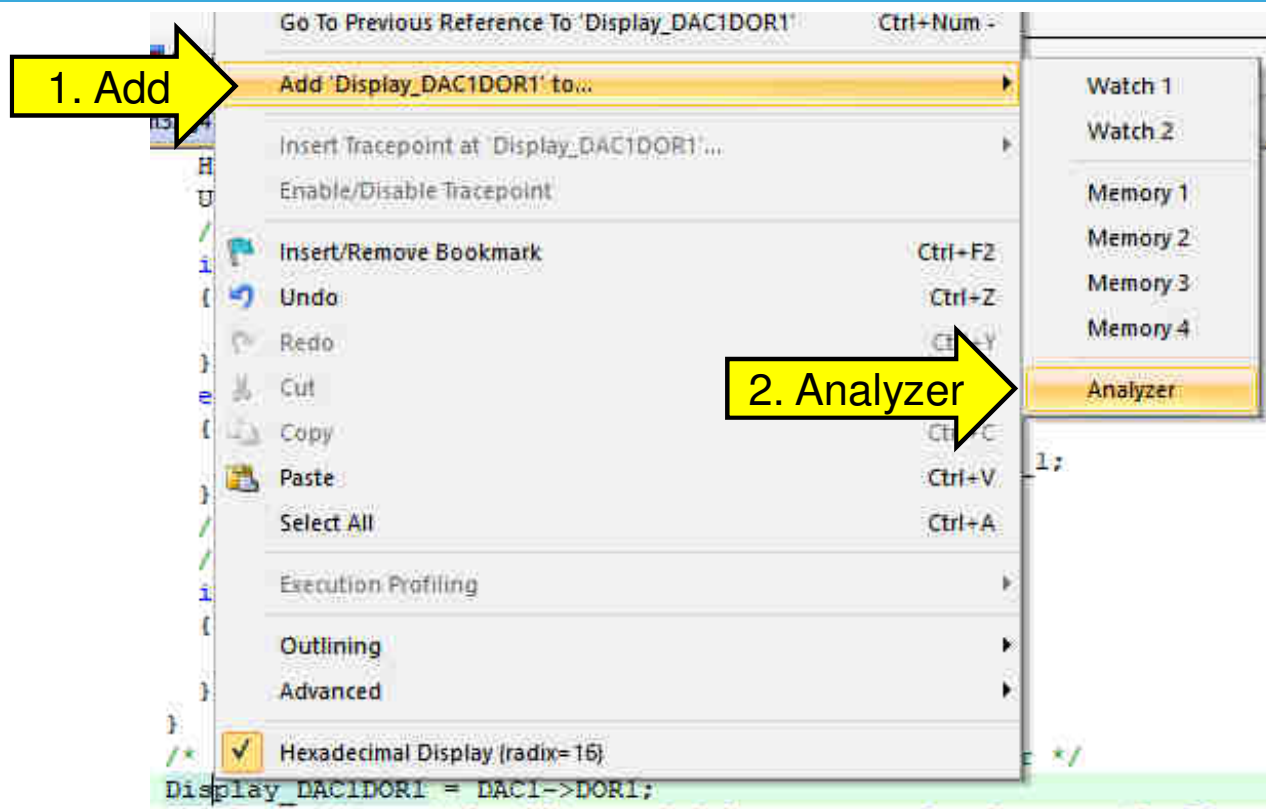


Display_DAC1DOR1 in the Logic Analyzer

```
174     Error_Handler();
175 }
176 }
177 /* Display the global variables in the Logic Analyzer */
178 Display_DAC1DOR1 = DAC1->DOR1;
179 HAL_Delay(1);    //1 millisecond delay to transfer data to the PC
180 Display_DAC3DOR2 = DAC3->DOR1;
181 HAL_Delay(1);    //1 millisecond delay to transfer data to the PC
182 Display_ADC3IN1 = ADC3->DR;
183 HAL_Delay(1);    //1 millisecond delay to transfer data to the PC
184 }
185 /* USER CODE END WHILE */
```

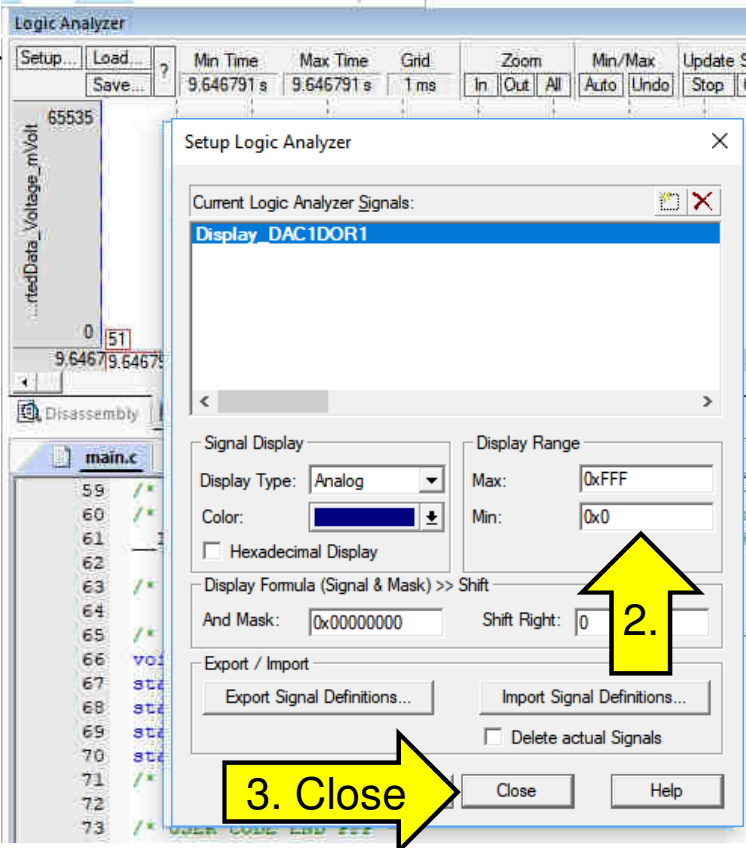
Right Click

Display_DAC1DOR1 in the Logic Analyzer



Setup – Min=0x0, Max = 0xFFFF

1. Setup



2.

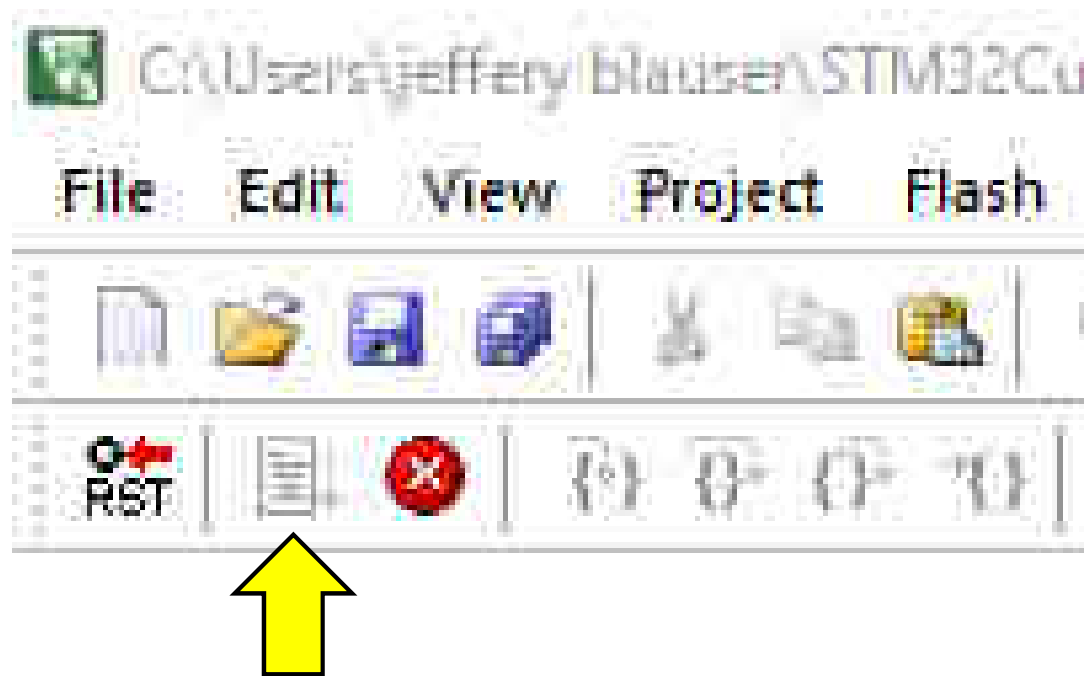
3. Close



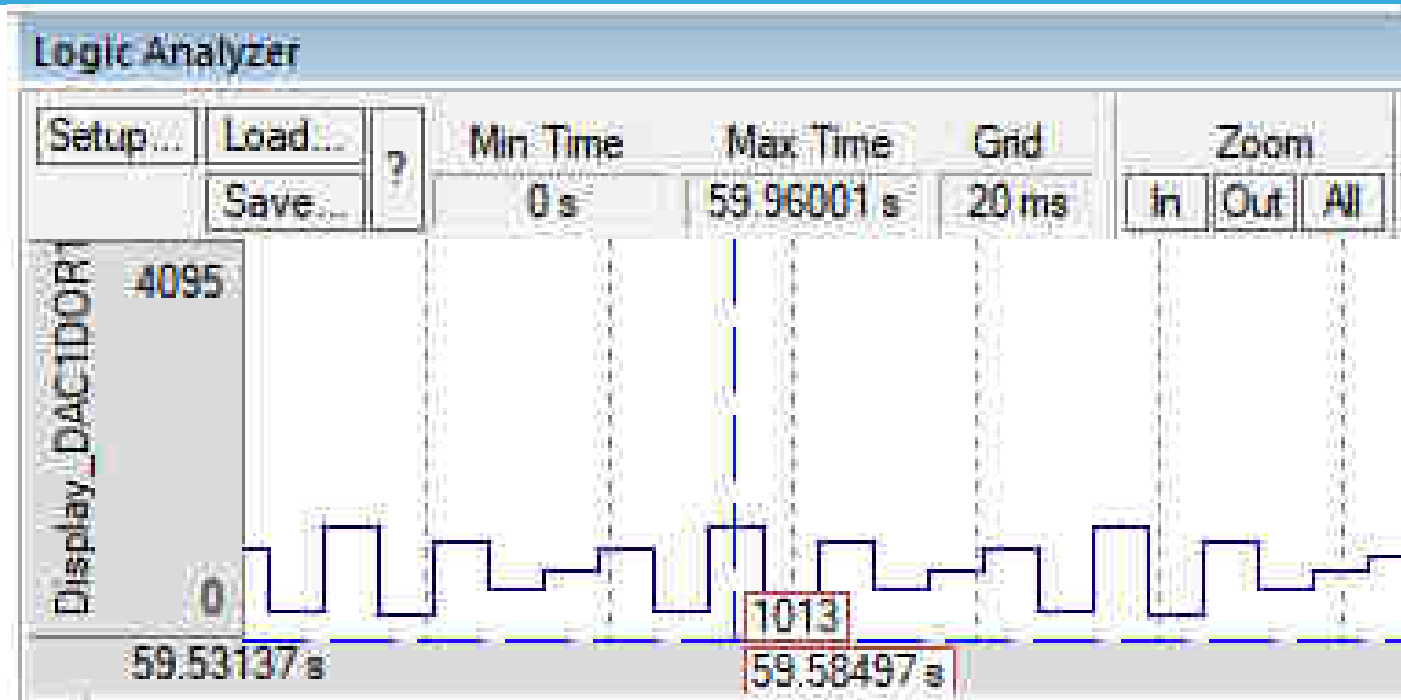
Keil MDK-ARM

180

RUN



DAC1_OUTPUT1

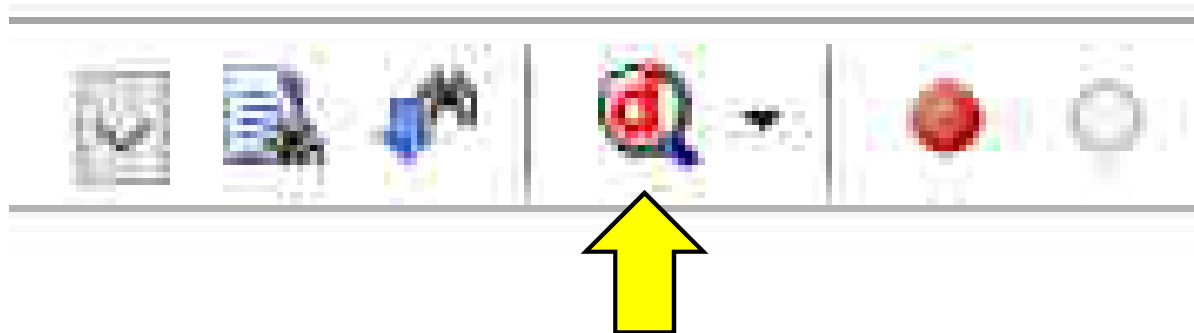
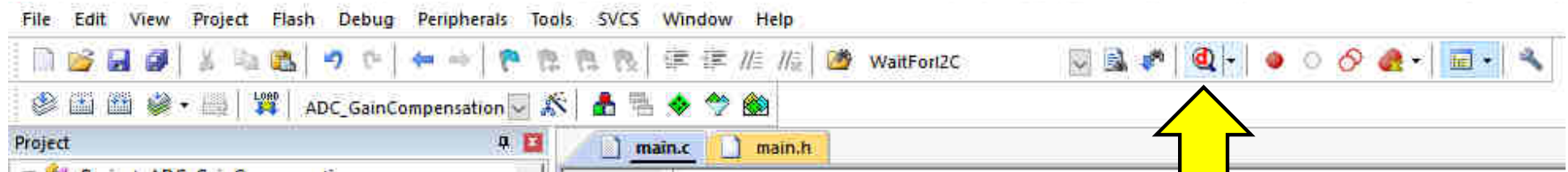




Keil MDK-ARM

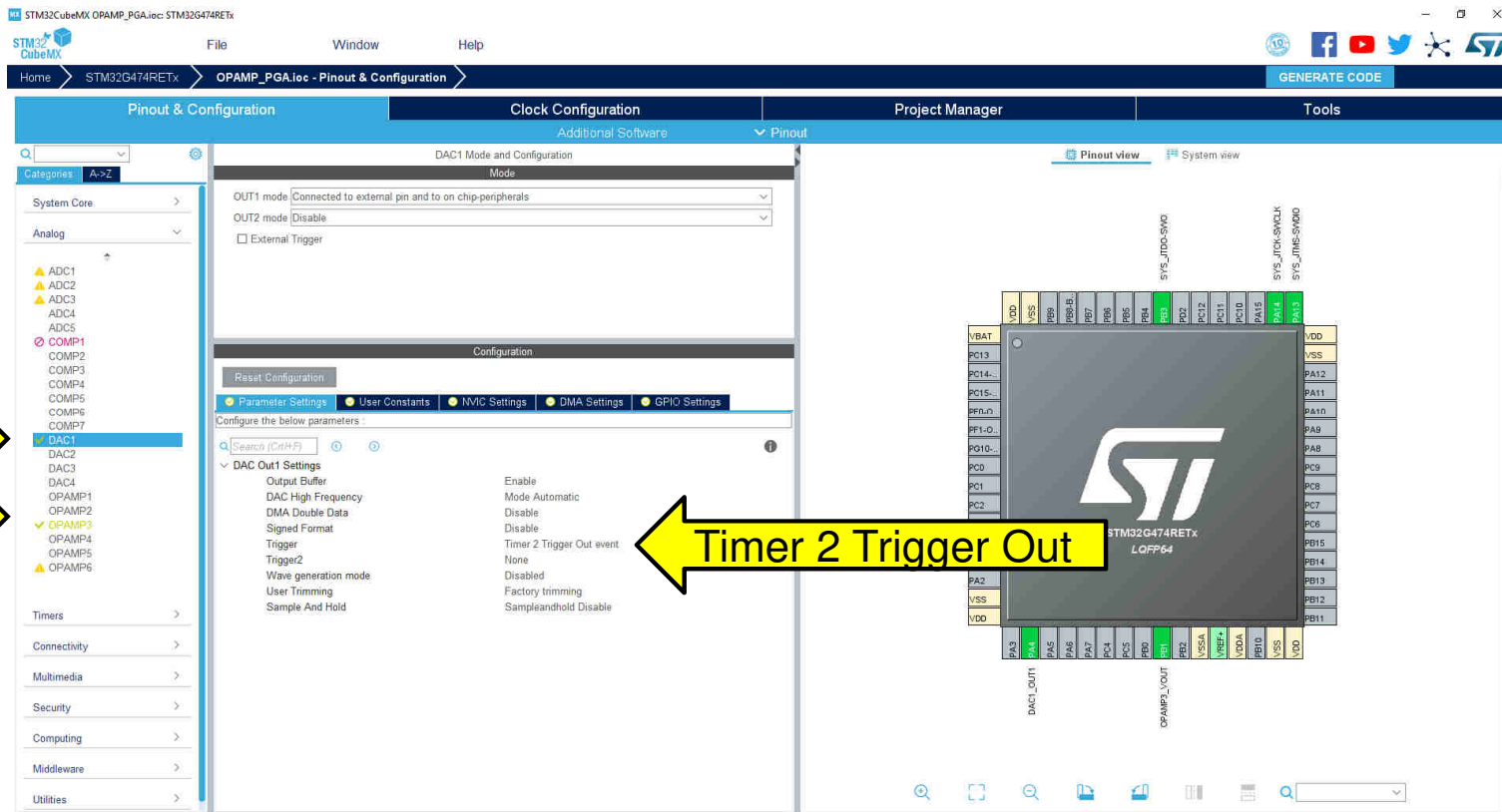
182

Debugger – Exit back to the Editor





Recall we are using, Op Amp 3





We will go with this for now, but where is the output?

The screenshot shows the STM32CubeMx software interface. On the left, the 'Categories' pane is expanded to 'Analog', showing a list of components: ADC1, ADC2, ADC3, ADC4, ADC5, COMP1 (disabled), COMP2, COMP3, COMP4, COMP5, COMP6, COMP7, DAC1 (checked), DAC2, DAC3, DAC4, OPAMP1, OPAMP2, and OPAMP3 (checked). A yellow arrow labeled 'OpAmp3' points to OPAMP3. On the right, the 'Mode' section shows 'Mode: PGA Not Connected', with checkboxes for 'Timer Controlled Mux Mode' (unchecked), 'Switching On Inverting Inputs' (Disable), and 'Switching On Non Inverting Inputs' (Disable). Below this is the 'Configuration' section with a 'Reset Configuration' button and three tabs: 'Parameter Settings' (checked), 'User Constants', and 'GPIO Settings'. Under 'Parameter Settings', the 'Basic Parameters' section is expanded, showing 'Power Mode' (Normal), 'PGA Gain' (2 or -1), and 'User Trimming' (Disable).



We will go with this for now, but where is the output?



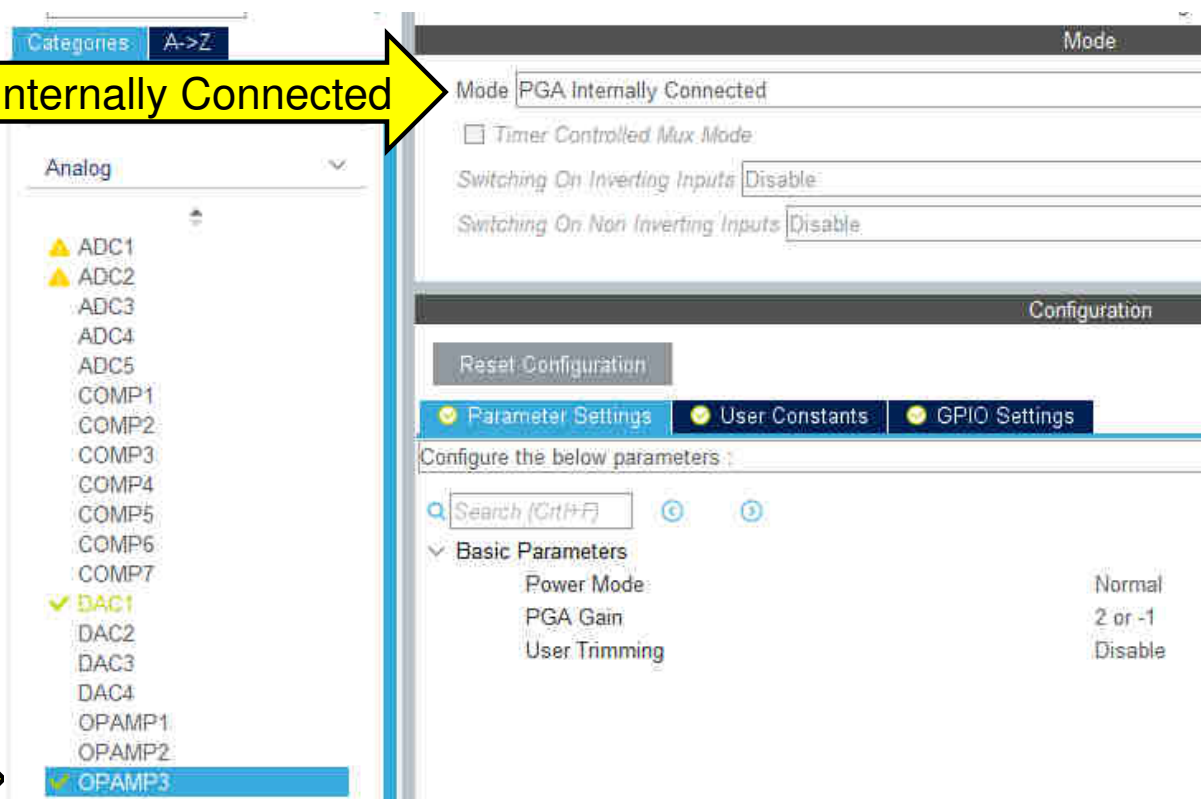
From the readme.txt file in the project, “Connect an oscilloscope to (OPAMP3_OUT, PB1: pin 24 on connector CN10) pin to display waveform”





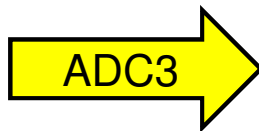
Set the PGA to an internal connection

2. PGA Internally Connected





Configure ADC3 – VOPAMP3, etc.



Categories: A-Z

System Core >

Analog >

- ADC1
- ADC2
- ADC3**
- ADC4
- ADC5
- COMP1
- COMP2
- COMP3
- COMP4
- COMP5
- COMP6
- COMP7
- DAC1
- DAC2
- DAC3
- DAC4
- OPAMP1
- OPAMP2
- OPAMP3**
- OPAMP4
- OPAMP5
- OPAMP6

Timers >

Connectivity >

Multimedia >

Security >

Computing >

Mode

- ☒ IN1 Single-ended
- ☐ IN5 Single-ended
- ☐ IN12 Single-ended
- ☐ Vbat Channel
- ☐ Vrefint Channel
- ☒ VOPAMP3 Channel

Configuration

Reset Configuration

Parameter Settings | User Constants | NVIC Settings | DMA Settings | GPIO Settings

Configure the below parameters :

Search (Ctrl+F)

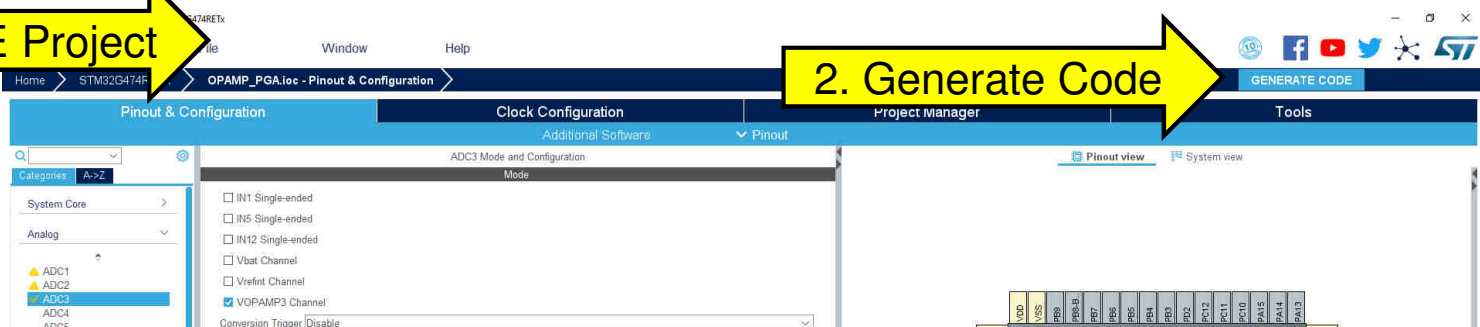
Clock Prescaler	Asynchronous clock mode divided by 4
Resolution	ADC 12-bit resolution
Data Alignment	Right alignment
Gain Compensation	0
Scan Conversion Mode	Disabled
End Of Conversion Selection	End of single conversion
Low Power Auto Wait	Disabled
Continuous Conversion Mode	Disabled
Discontinuous Conversion Mode	Disabled
DMA Continuous Requests	Disabled
Overrun behaviour	Overrun data overwritten
ADC_Regular_ConversionMode	
Enable Regular Conversions	Enable
Enable Regular Oversampling	Disable
Number Of Conversion	1
External Trigger Conversion Source	Timer 2 Trigger Out event
External Trigger Conversion Edge	Trigger detection on the falling edge
Rank	1
Channel	Channel Vopamp3
Sampling Time	2.5 Cycles



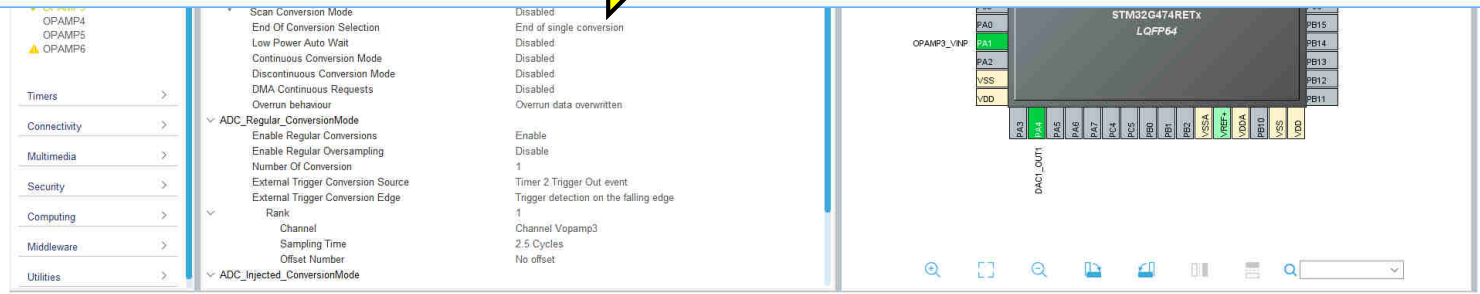
File – Save Project – Generate Code

1. File – SAVE Project

2. Generate Code



3. Do not Click – Go to Keil

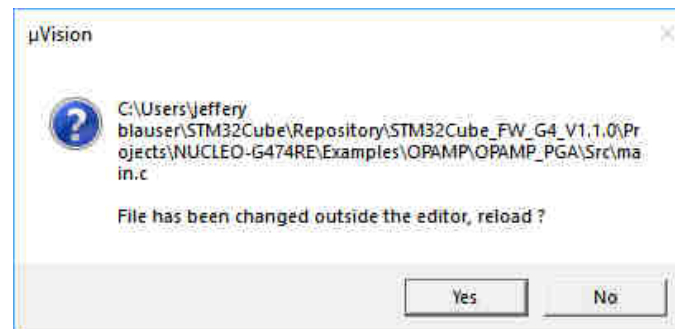




Keil MDK-ARM

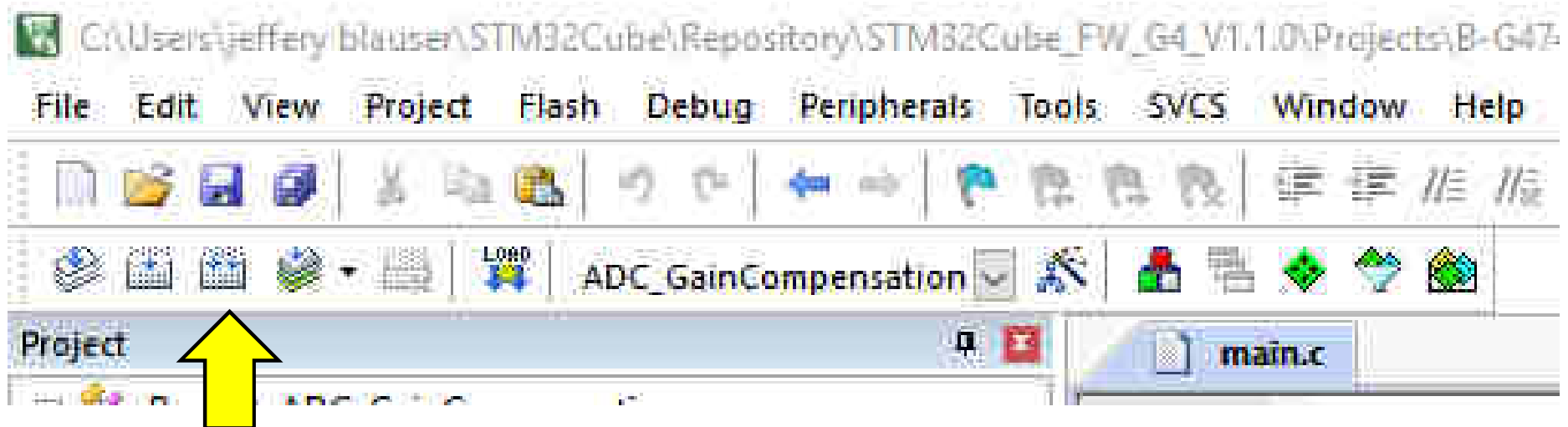
189

Click YES 3x





Build



Build Output

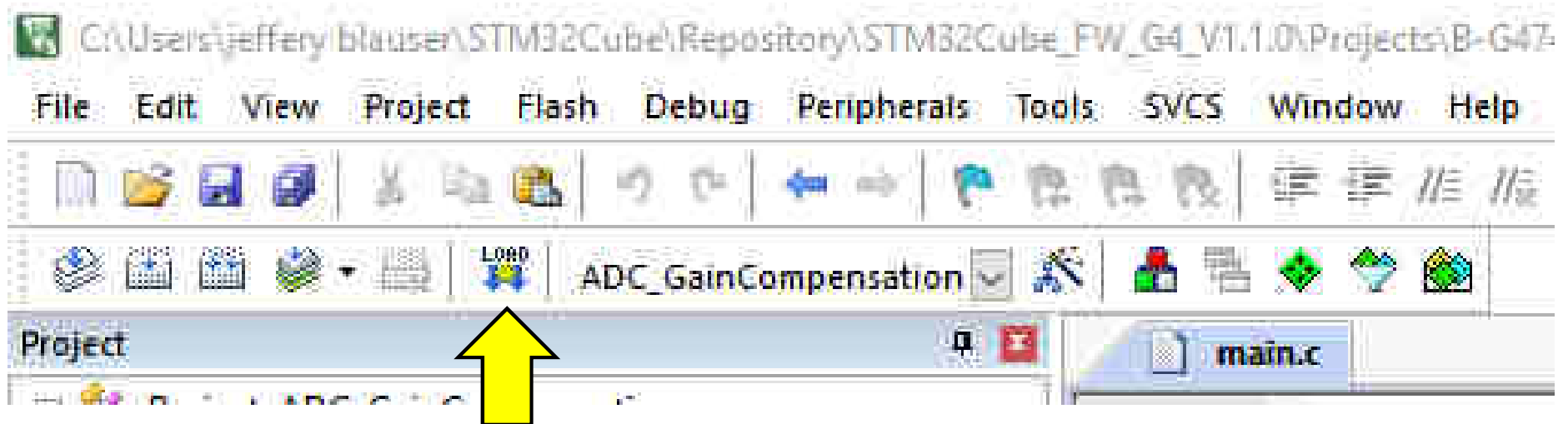
```
Program Size: Code=6516 RO-data=596 RW-data=44 ZI-data=1268
FromELF: creating hex file...
OPAMP_PGA\OPAMP_PGA.hex: Warning: Q9931W: Your license for feature Keil will expire in 30 days
Finished: 0 information, 1 warning and 0 error messages.
"OPAMP_PGA\OPAMP_PGA.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:01:22
```



Keil MDK-ARM

191

Load

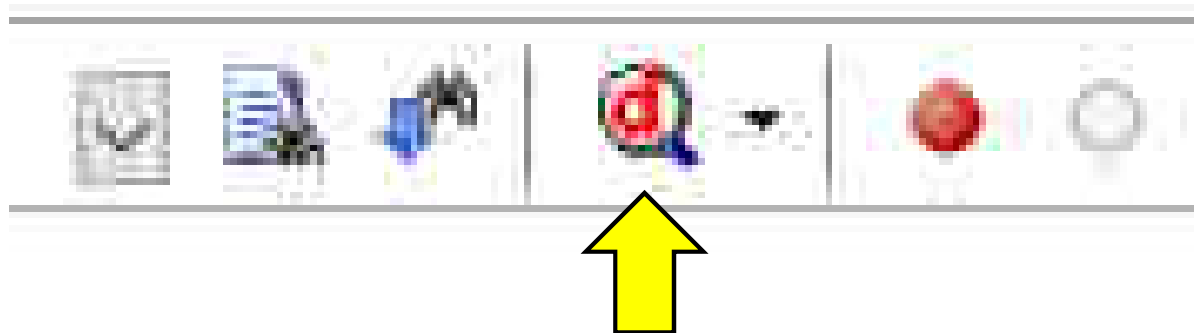
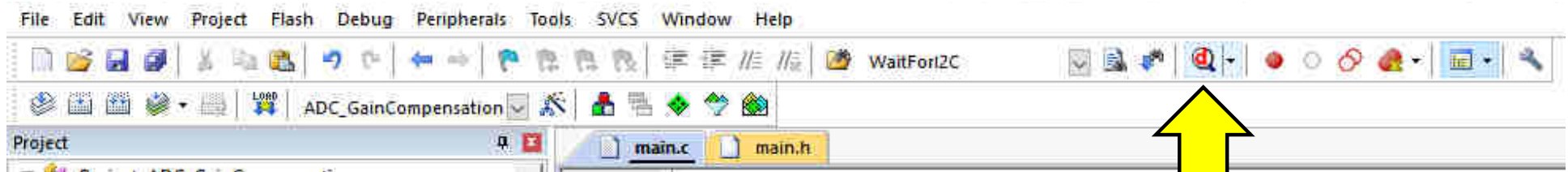




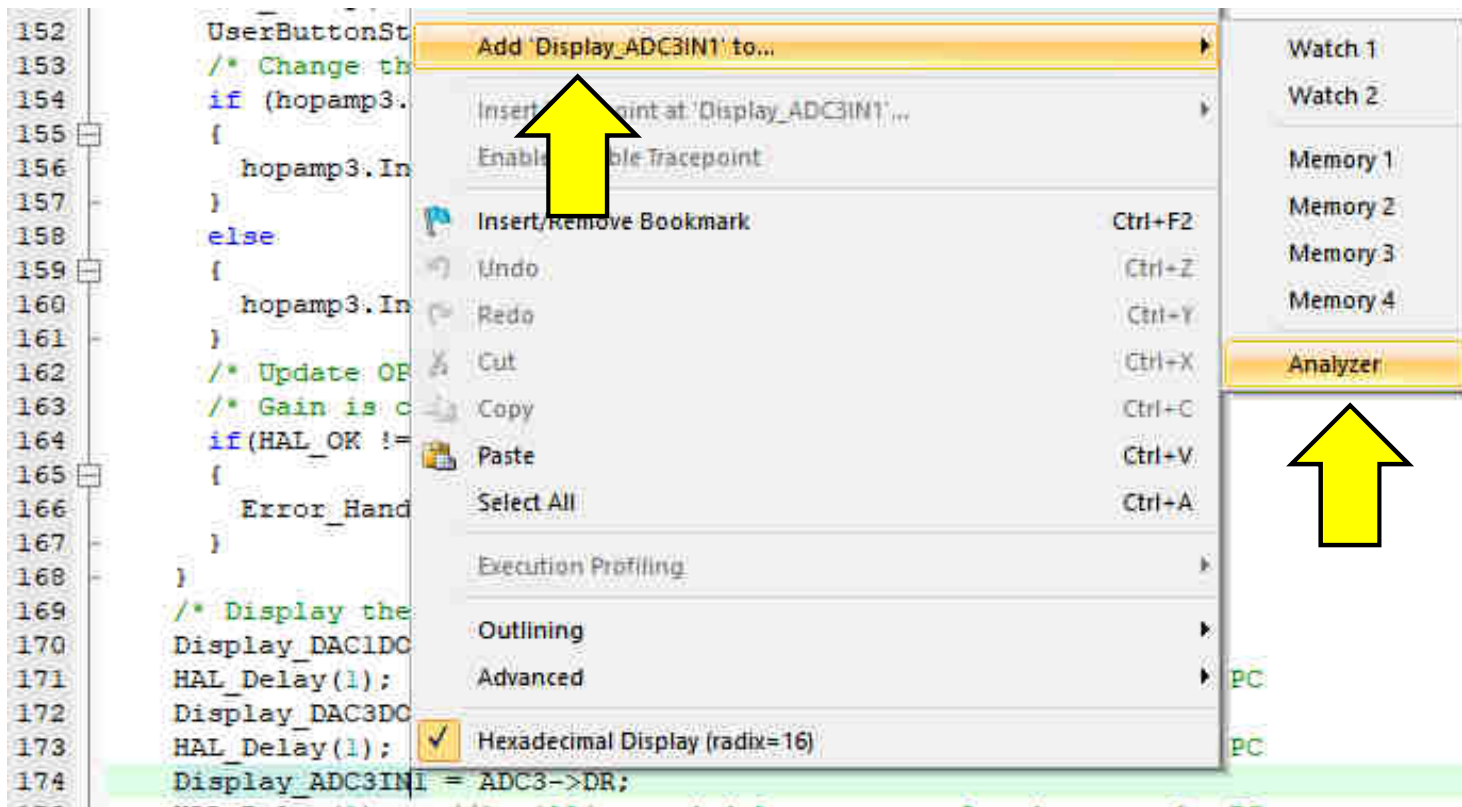
Keil MDK-ARM

192

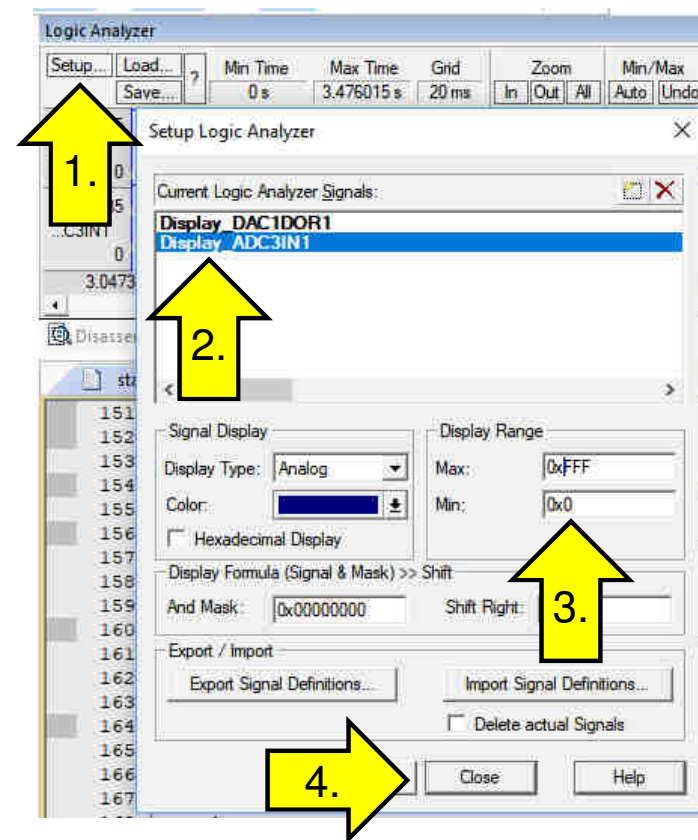
Start Debugger



Add Display_ADC3IN1 to the Logic Analyzer

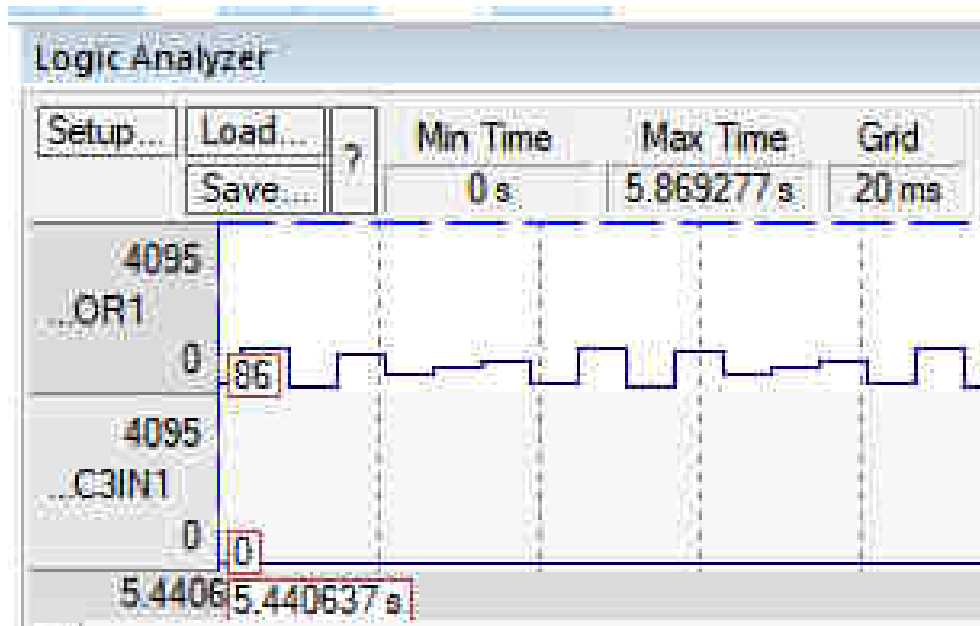


SETUP – ADC3IN1 – 0 to 0xFFFF - Close





DAC output is good, Why is the ADC zero?

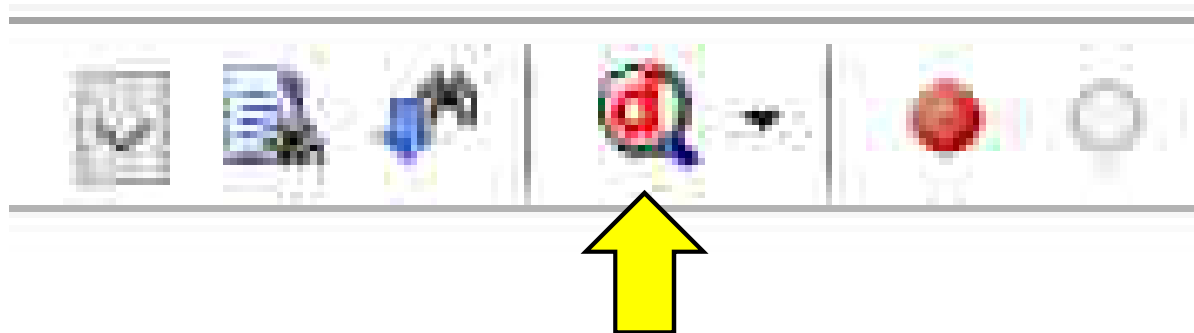
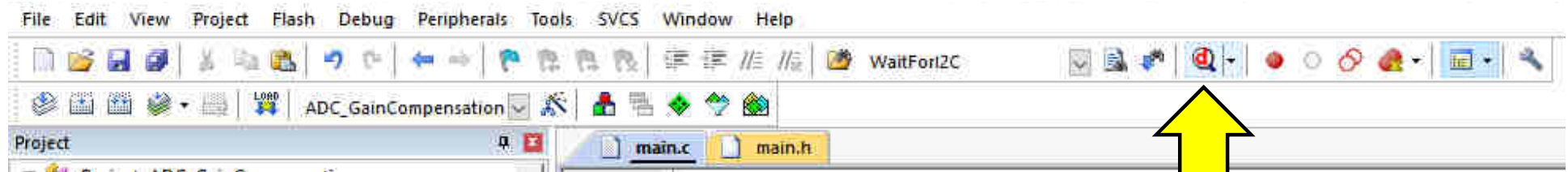




Keil MDK-ARM

196

Exit Debugger





DAC3 – OUT2 – Timer2 Trigger Out Event

Categories A-Z

System Core >

Analog v

- ADC1
- ADC2
- ✓ ADC3
- ADC4
- ADC5
- COMP1
- COMP2
- COMP3
- COMP4
- COMP5
- COMP6
- COMP7
- ✓ DAC1
- DAC2
- ✓ DAC3
- DAC4
- OPAMP1
- OPAMP2
- ✓ OPAMP3
- OPAMP4
- OPAMP5
- OPAMP6

DAC3

Mode

- ☐ OUT1 Connected to on chip-peripherals only
- ☒ OUT2 Connected to on chip-peripherals only
- ☐ External Trigger

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings DMA Settings

Configure the below parameters :

Search (Ctrl+F)

✓ DAC Out2 Settings

Output Buffer	Disable
DAC High Frequency	Mode Automatic
DMA Double Data	Disable
Signed Format	Disable
Trigger	Timer 2 Trigger Out event
Trigger2	None
Wave generation mode	Triangle wave generation
Maximum Triangle Amplitude	1023
User Trimming	Factory trimming
Sample And Hold	Sampleandhold Disable



DAC3 – DMA - ADD - DMA1_CH2 – Periph=word / Mem=half word

The screenshot shows the STM32CubeMx configuration interface. On the left, the 'Analog' category is expanded, and 'DAC3' is selected. A yellow arrow points from the text 'DAC3' to this selection. The main panel shows the 'DMA Settings' tab, which includes a table of DMA requests and a settings section below.

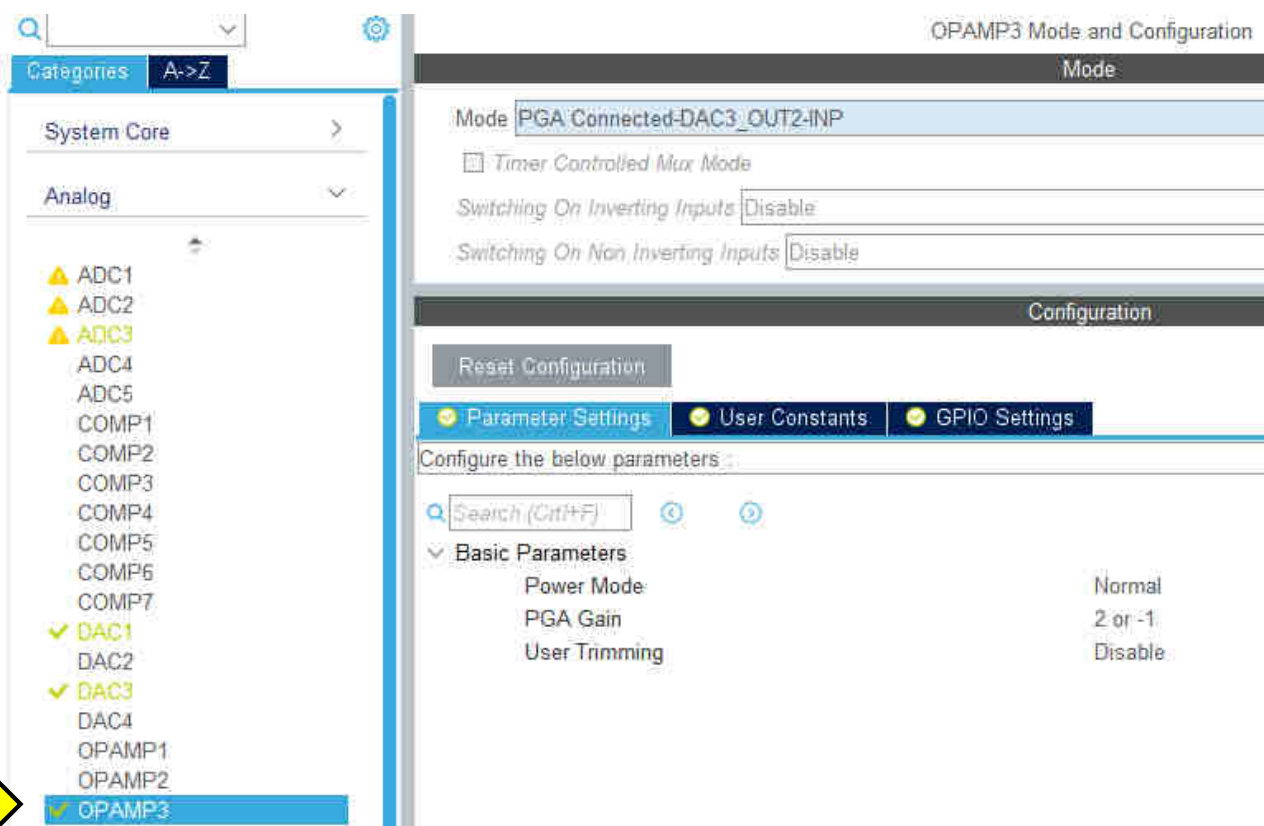
DMA Request	Channel	Direction	Priority
DAC3_CH2	DMA1 Channel 2	Memory To Peripheral	Medium

Buttons: Add, Delete

DMA Request Settings:

Peripheral		Memory
Mode: Circular	Increment Address: ☐	☒
Data Width: Word		Half Word

OpAmp3 Mode – PGA connected DAC3_OUT2-INP



The screenshot displays the STM32CubeMx configuration interface. On the left, the 'Analog' category is expanded, showing a list of components including ADC1 through ADC5, COMP1 through COMP7, DAC1 through DAC4, and OPAMP1 through OPAMP3. OPAMP3 is selected and highlighted in blue. A yellow arrow points from the 'OpAmp3' text to the selected OPAMP3 component.

The main configuration area is titled 'OPAMP3 Mode and Configuration'. It shows the 'Mode' set to 'PGA Connected-DAC3_OUT2-INP'. Below this, there are checkboxes for 'Timer Controlled Mux Mode' (unchecked), 'Switching On Inverting Inputs' (Disable), and 'Switching On Non Inverting Inputs' (Disable).

The 'Configuration' section includes a 'Reset Configuration' button and three tabs: 'Parameter Settings' (selected), 'User Constants', and 'GPIO Settings'. Under 'Parameter Settings', the 'Basic Parameters' are listed:

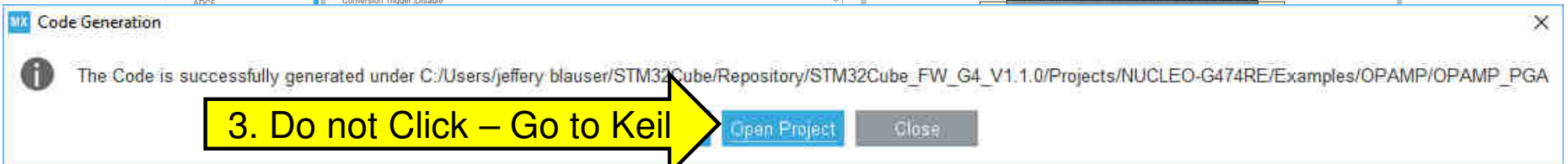
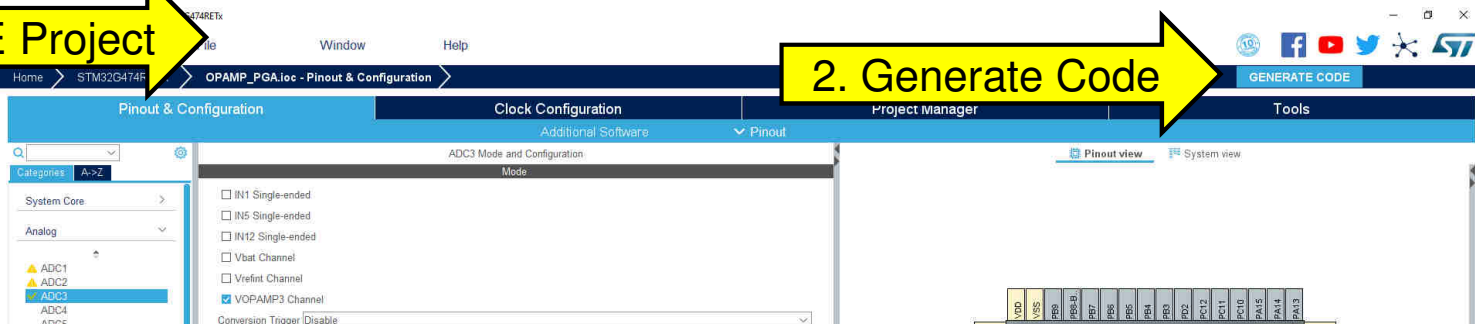
- Power Mode: Normal
- PGA Gain: 2 or -1
- User Trimming: Disable



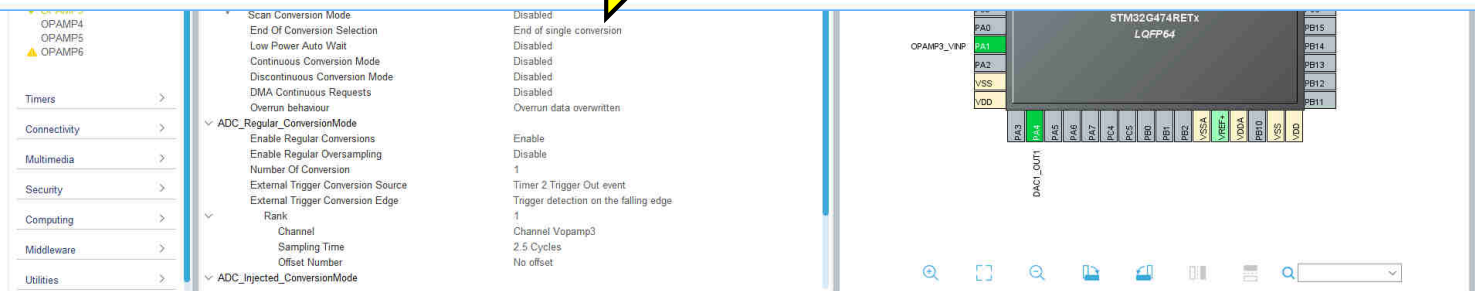
File – Save Project – Generate Code

1. File – SAVE Project

2. Generate Code



3. Do not Click – Go to Keil

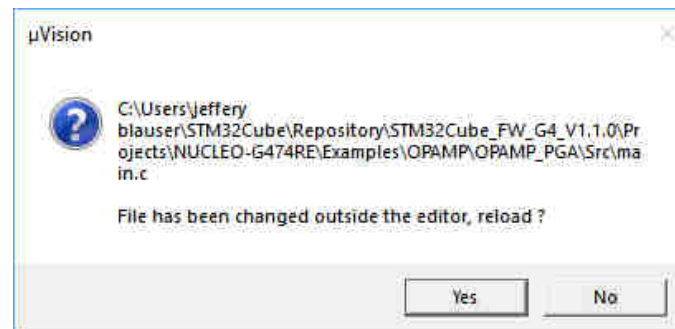




Keil MDK-ARM

201

Click YES 3x



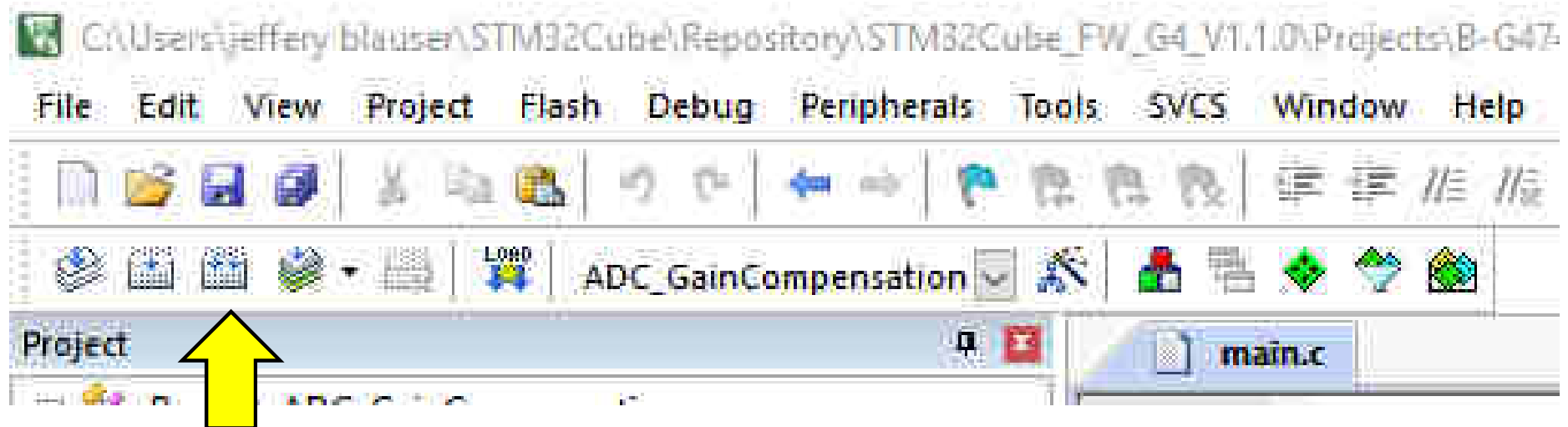


In Main.c, change the direction of the DMA to DAC3_OUT2

```
129  /* Infinite loop */
130  /* USER CODE BEGIN WHILE */
131  /*##- Enable DAC Channel and associated DMA #####*/
132  if(HAL_OK != HAL_DAC_Start_DMA(&hdac3, DAC_CHANNEL_2,
133                                  (uint32_t*)Sinel2bit, 32, DAC_ALIGN_12B_R))
134  ,
```



File-Save-Build



Build Output

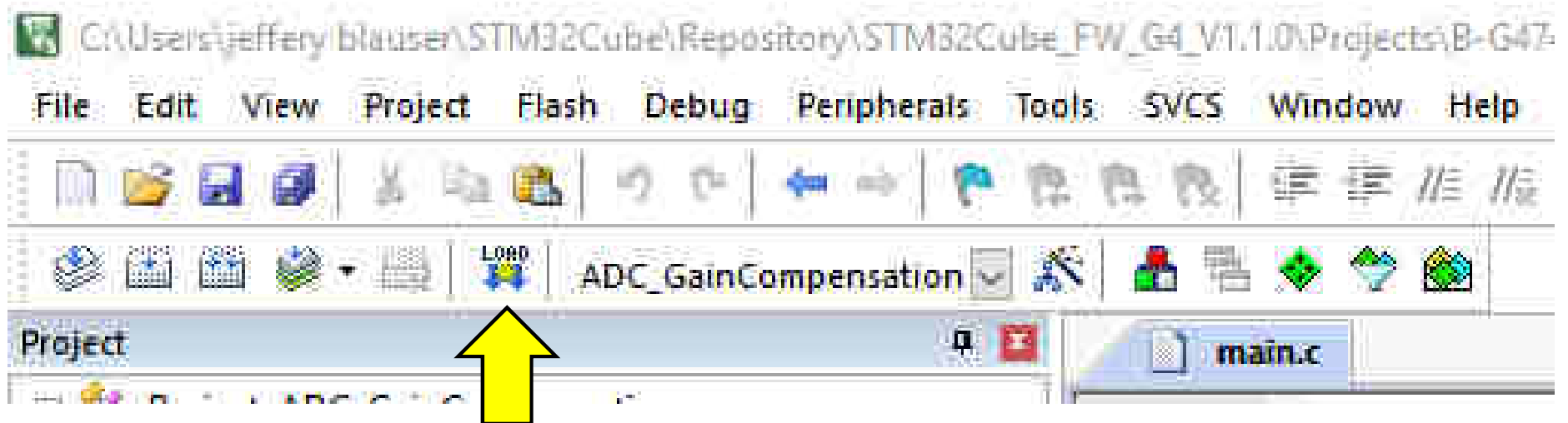
```
Program Size: Code=6516 RO-data=596 RW-data=44 ZI-data=1268  
FromELF: creating hex file...  
OPAMP_PGA\OPAMP_PGA.hex: Warning: Q9931W: Your license for feature Keil will expire in 30 days  
Finished: 0 information, 1 warning and 0 error messages.  
"OPAMP_PGA\OPAMP_PGA.axf" - 0 Error(s), 0 Warning(s).  
Build Time Elapsed: 00:01:22
```



Keil MDK-ARM

204

Load

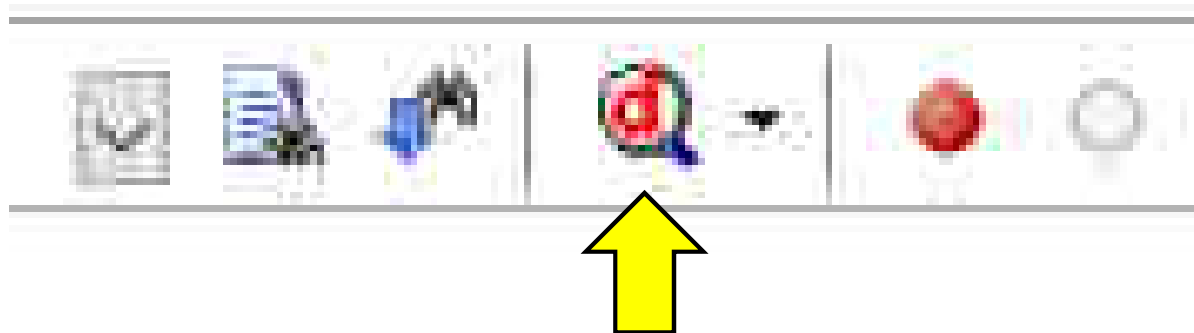
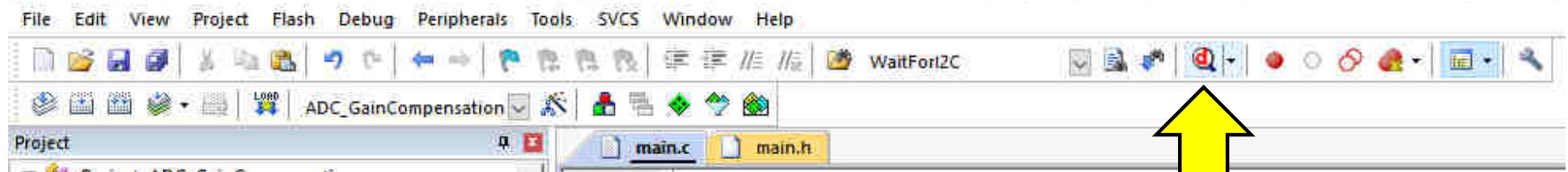




Keil MDK-ARM

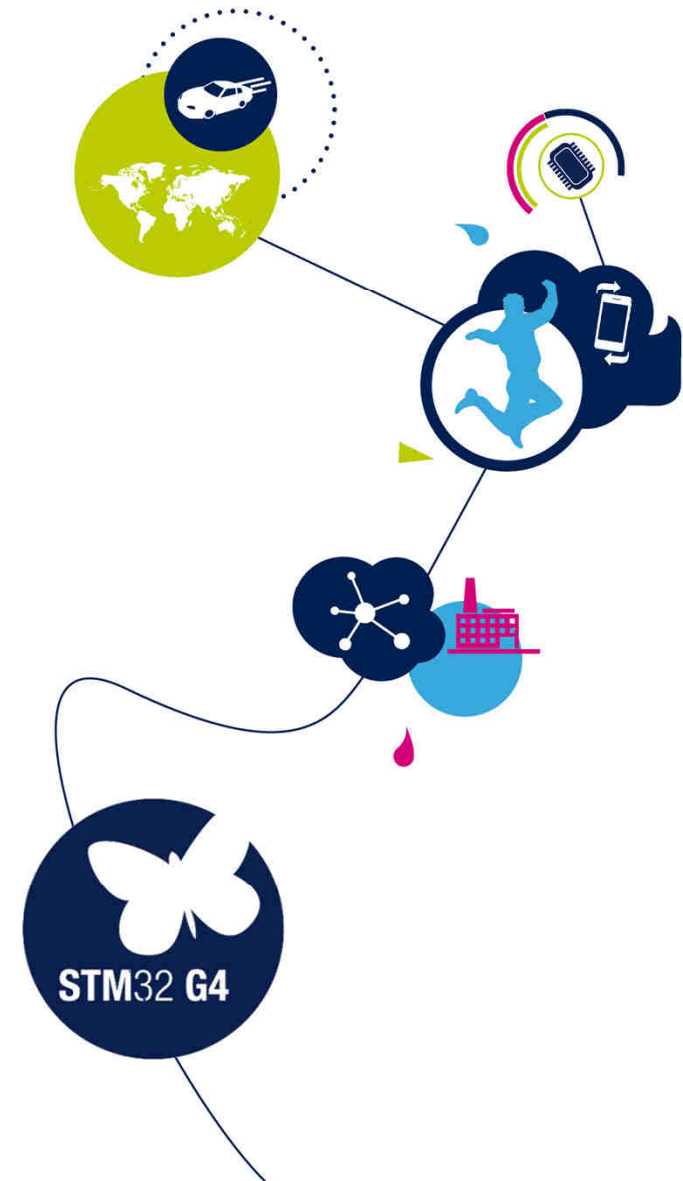
205

Start Debugger



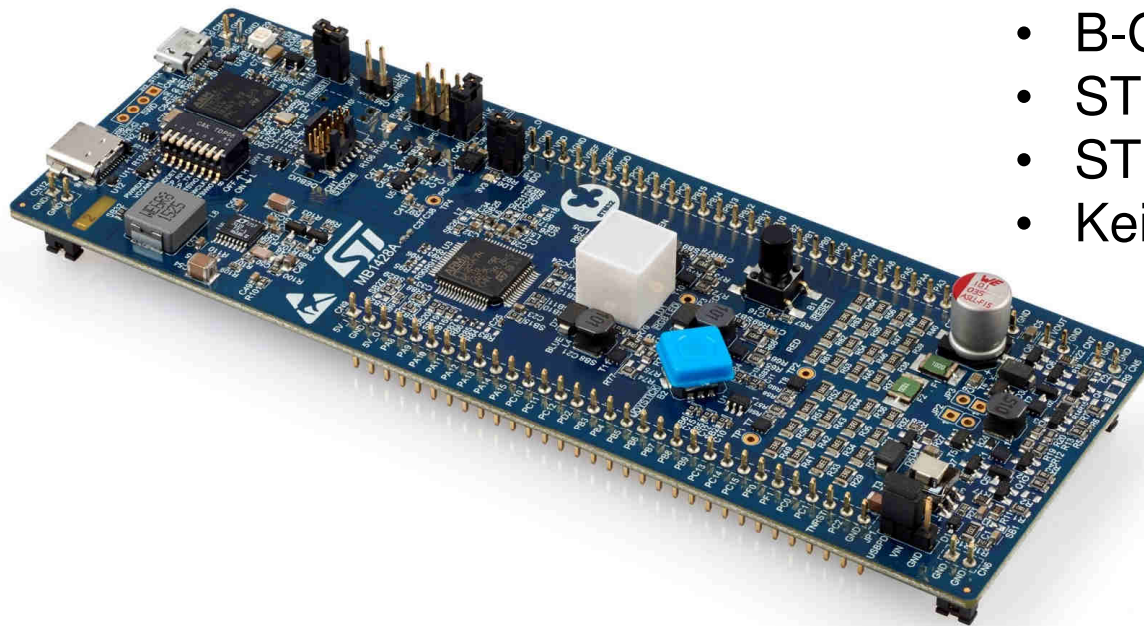
STM32G4 mixed-signal MCU workshop

*Hands On:
High Resolution Timer (HRTIM)*





Systems Check



- USB cable + USB-C Cable
- B-G474E-DPOW1 Discovery Board
- STM32G4G4 Family Cube Library
- STM32CubeMx
- Keil MDK-ARM



High Resolution Timer

208

G4 HRTIM

- Generates up to 12 outputs
- Up to 6 complementary pair PWM outputs
- 184pS resolution
- 6 sub timers, 4 compares each
- 32 set/reset sources for 12 outputs
- Sensitive to up to 10 external events
- Deadtime instertion down to 735pS
- Burst mode
- 8 interrupt vectors – 14 sources each

Internally connected to:

- DMA
- ADC's
- DAC's
- Comparators
- Timers
- NVIC



High Resolution Timer

209

G4 HRTIM – What's new?

The timer F unit has been added to have a total of 12 outputs available, together with a 6th FAULT input

Rational: possibility to address larger converters

- Triple-interleaved half-bridge LLC, requiring 3 x 4 outputs

 - 2x at primary for half-bridge control and 2x at secondary for synchronous rectification

- Dual interleaved full-bridge LLC (2x 6 outputs)

- Dual phase-shift full bridge converter (2x 6 outputs)

Implementation

- Timer F new control / status bits have been added in reserved areas, to maintain partial compatibility whenever possible

- Still, as many registers were full, the option was chosen to maintain all bits in single 32-bit registers, but breaking compatibility

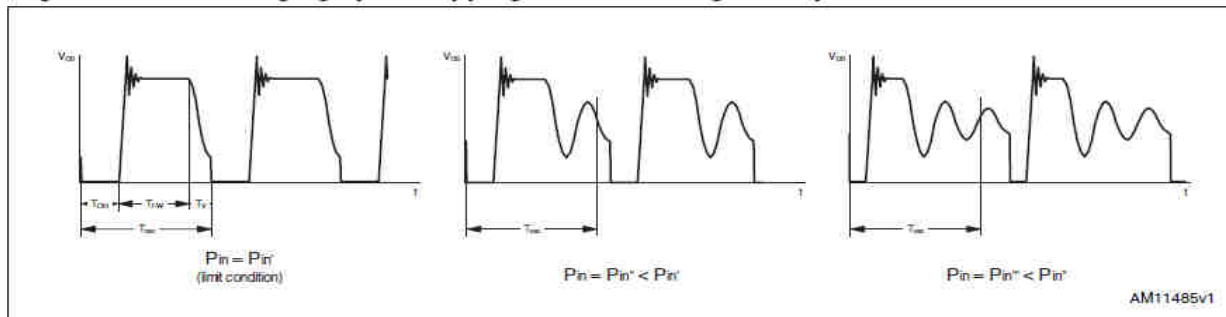
- When bit position was changed, this is managed transparently by the HAL / LL



G4 HRTIM – What's new? – External Event counter 1 of 2

- The HRTIM v1 offers many options for timing-driven filtering (blanking and windowing)
 - Typical use case: leading edge blanking (LEB)
- The HRTIM v2 features event-based filtering: an external event must have occurred a given number of times before being considered valid
 - Typical use case: valley skipping mode for flyback converters
 - At low-load, the next PWM cycle must start after N cycles of ringing (the lower the load, the higher N)

Figure 9. Drain ringing cycle skipping as the load is gradually reduced



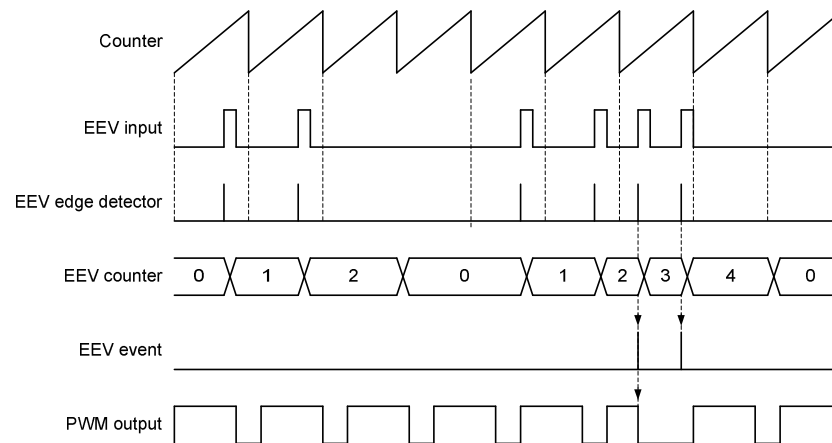


High Resolution Timer

211

G4 HRTIM – What's new? – External Event counter 2 of 2

- **Immediate mode**
 - Event is generated as soon as N consecutive events are occurring during a PWM period
 - Event counter is reset on each and every new PWM period
- **Cumulative mode (figure below)**
 - event must occur at least once during multiple PWM periods
 - counter is reset only if the event did not appear during the last PWM period





High Resolution Timer

212

Lab steps 1 of 4

- This Hands On example shows how to configure the HRTIM to control a non-inverting buck-boost converter timer.
- This example uses the STM32G4xx HRTIM HAL and LL APIs.
- You will use the timer unit C and D and TC1/TC2/TD1/TD2 outputs (respectively PB12, PB13, PB14, PB15).



High Resolution Timer

213

Lab steps 2 of 4

- The FAULT2 input is enabled on PA15 (active LO) to demonstrate PWM shut down (low level sensitive), for all outputs.
- When the fault is triggered (PA15 input connected to GND), TC1, TC2, TD1, TD2 signals are shut-down. The system can be re-armed by pressing the user button.
- ADC1 is configured to have conversions triggered in the middle of the converter ON time, on PA1 (BUCKBOOST_VIN) and PA3 (BUCKBOOST_VOUT) inputs.



High Resolution Timer

214

Lab steps 3 of 4

- To run the demo, the Vin input pin of the BUCK-BOOST converter must be connected to the 5V_O supply on the CN9 connector or on an external power supply. The resulting voltage is available on Vout pin
- LEDs are indicating the following:
 - **Green** LED4: blinks during BUCK operation
 - **Orange** LED3: blinks during BOOST operation
 - **Red** LED5: blinks when FAULT is triggered



High Resolution Timer

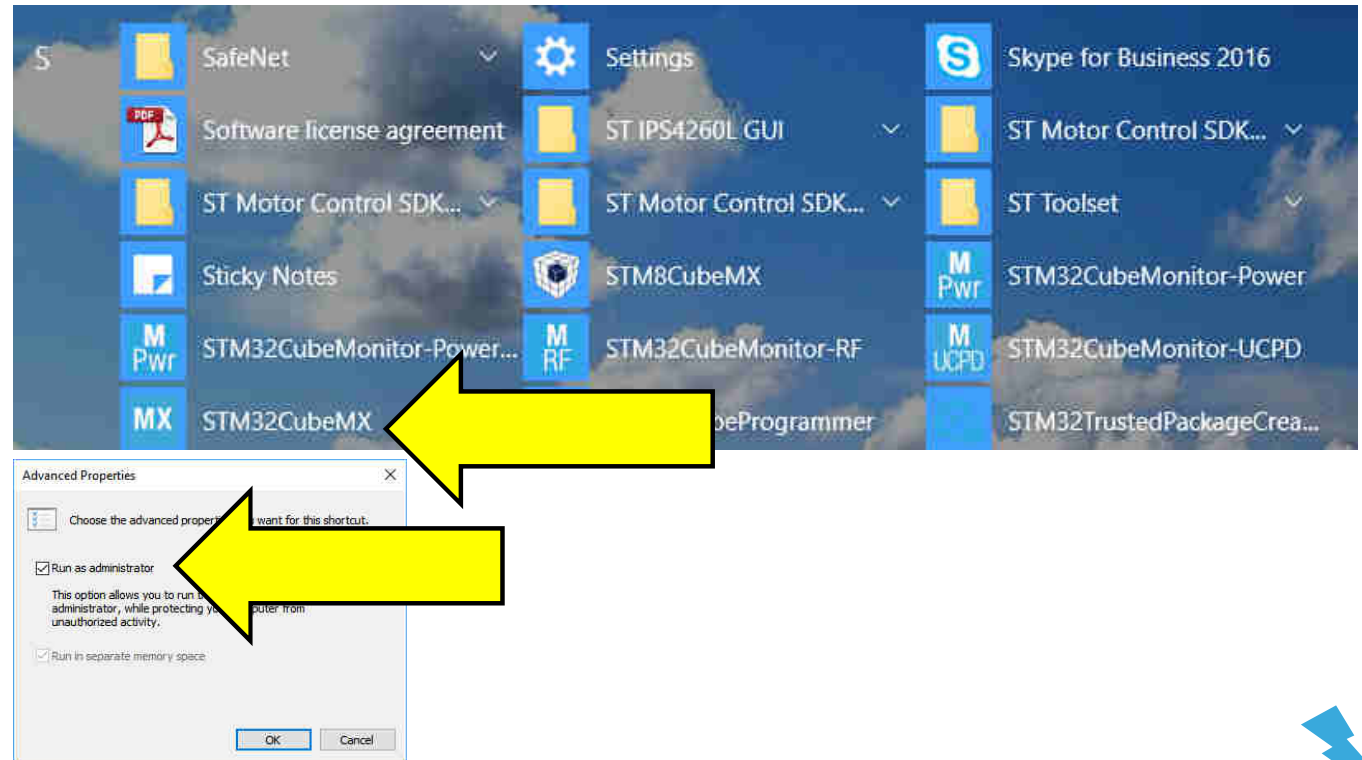
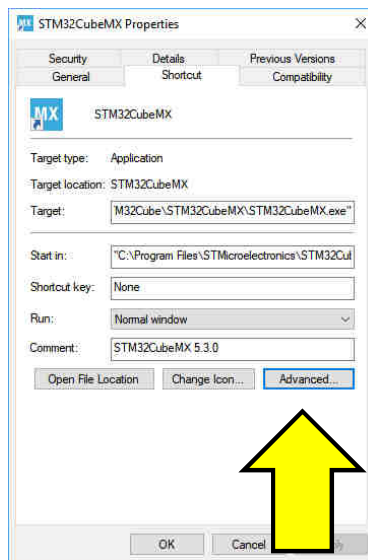
215

Lab steps 4 of 4

- The demo starts in BUCK mode, and the duty cycle is slowly adjusted in the TIMC IRQ handler to have V_{out} continuously varying below V_{in} value.
- If the push-button is pressed and the voltage is below 5V, the boost mode is enabled (this is to prevent exceeding the Discovery Kit max output voltage (15V)).
- The voltage is increased above V_{in} value (with a fixed duty cycle).
- If the push-button is pressed again, the output capacitor is first de-energized down to 2.5V before re-enabling the buck mode.



Start up





Splash Page – G4 Library Installed?

The image shows the STM32CubeMX splash page and the Embedded Software Packages Manager window. The splash page has a sidebar with navigation options: 'New project from MCU', 'New project from STBoard', and 'New project from Cross Selection'. The main area displays 'Manage software installations' with buttons for 'CHECK FOR UPDATES' and 'INSTALL / REMOVE'. A large yellow arrow points from the 'INSTALL / REMOVE' button to the text 'Install/Remove'. Below this is a section for 'New multicore STM32MP1 Series for Industrial and IoT applications' featuring an STM32MP1 chip and the OpenSTLinux Distribution logo.

The Embedded Software Packages Manager window is open, showing a table of installed packages. A yellow arrow points to the 'STM32G4' section.

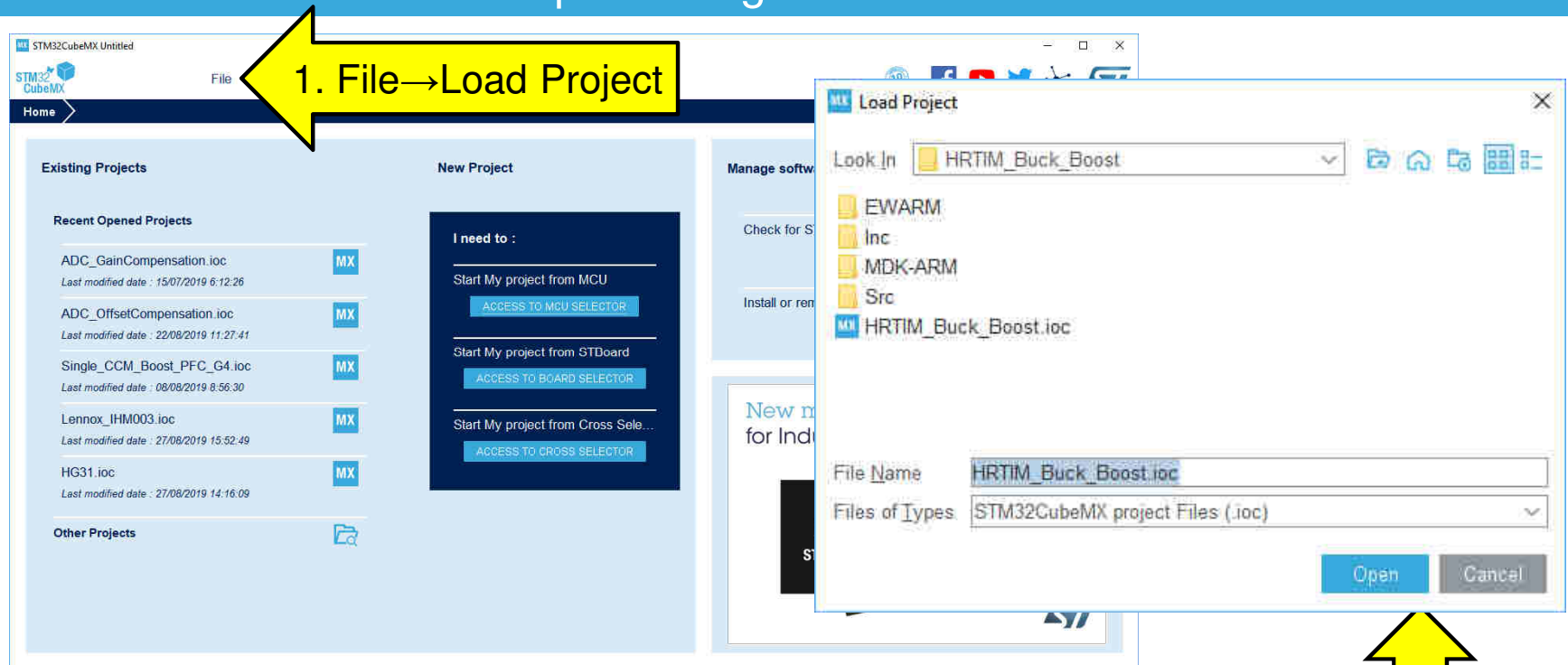
Description	Installation	Available Version
STM32F7		
STM32G0		
STM32G4		
STM32Cube MCU Package for STM32G4 Series	1.1.0	1.1.0
STM32Cube MCU Package for STM32G4 Series	1.0.1	1.0.1

Details

From Local From Url Refresh Install Now Remove Now Close

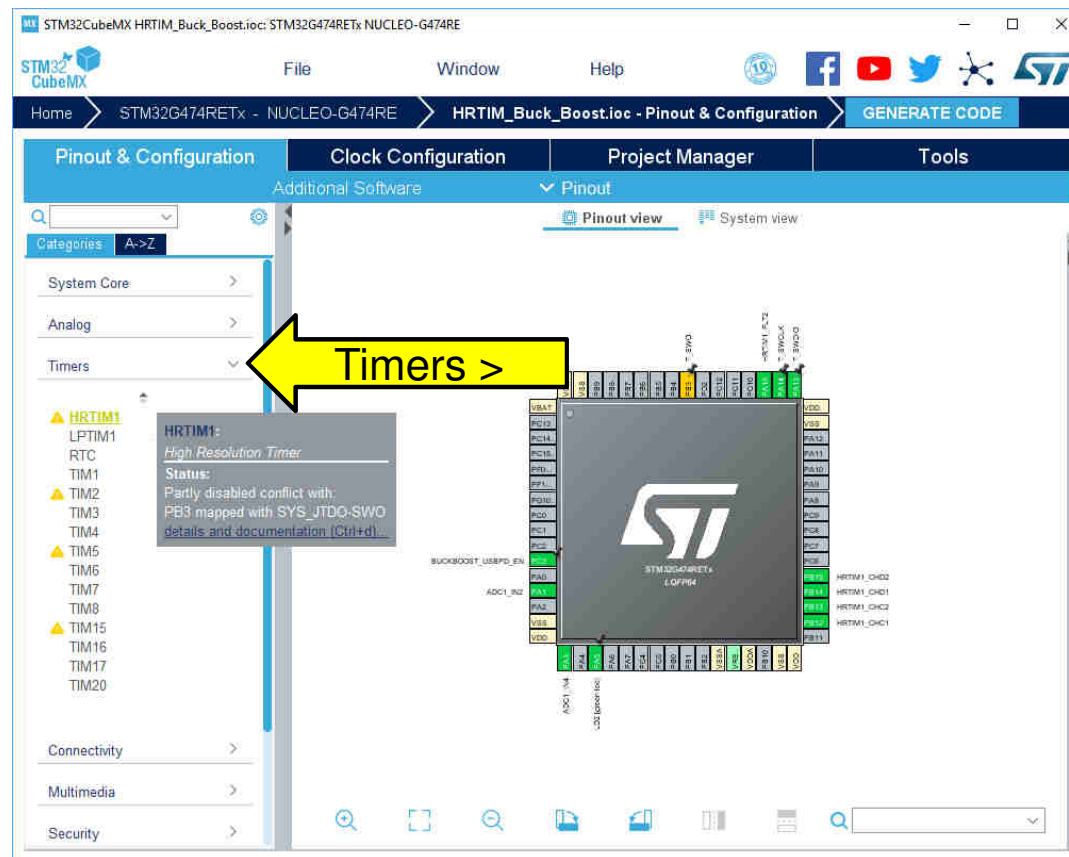


Splash Page

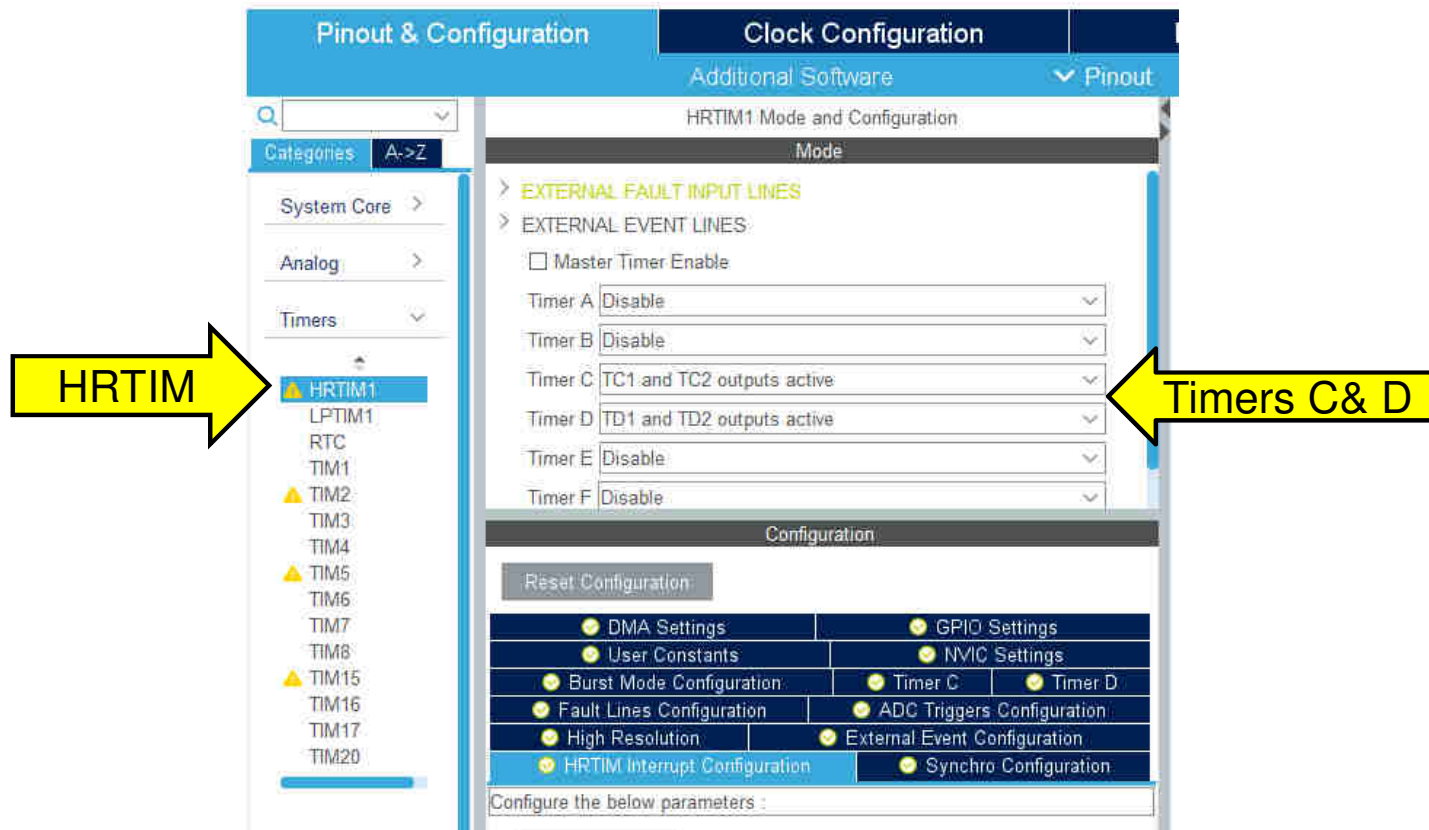




Graphic Design Representation



HRTIM Channels C & D



Pinout & Configuration

Categories A->Z

- System Core >
- Analog >
- Timers >
 - HRTIM1
 - LPTIM1
 - RTC
 - TIM1
 - TIM2
 - TIM3
 - TIM4
 - TIM5
 - TIM6
 - TIM7
 - TIM8
 - TIM15
 - TIM16
 - TIM17
 - TIM20

HRTIM1 Mode and Configuration

Mode

- > EXTERNAL FAULT INPUT LINES
- > EXTERNAL EVENT LINES
 - ☐ Master Timer Enable
 - Timer A Disable
 - Timer B Disable
 - Timer C TC1 and TC2 outputs active
 - Timer D TD1 and TD2 outputs active
 - Timer E Disable
 - Timer F Disable

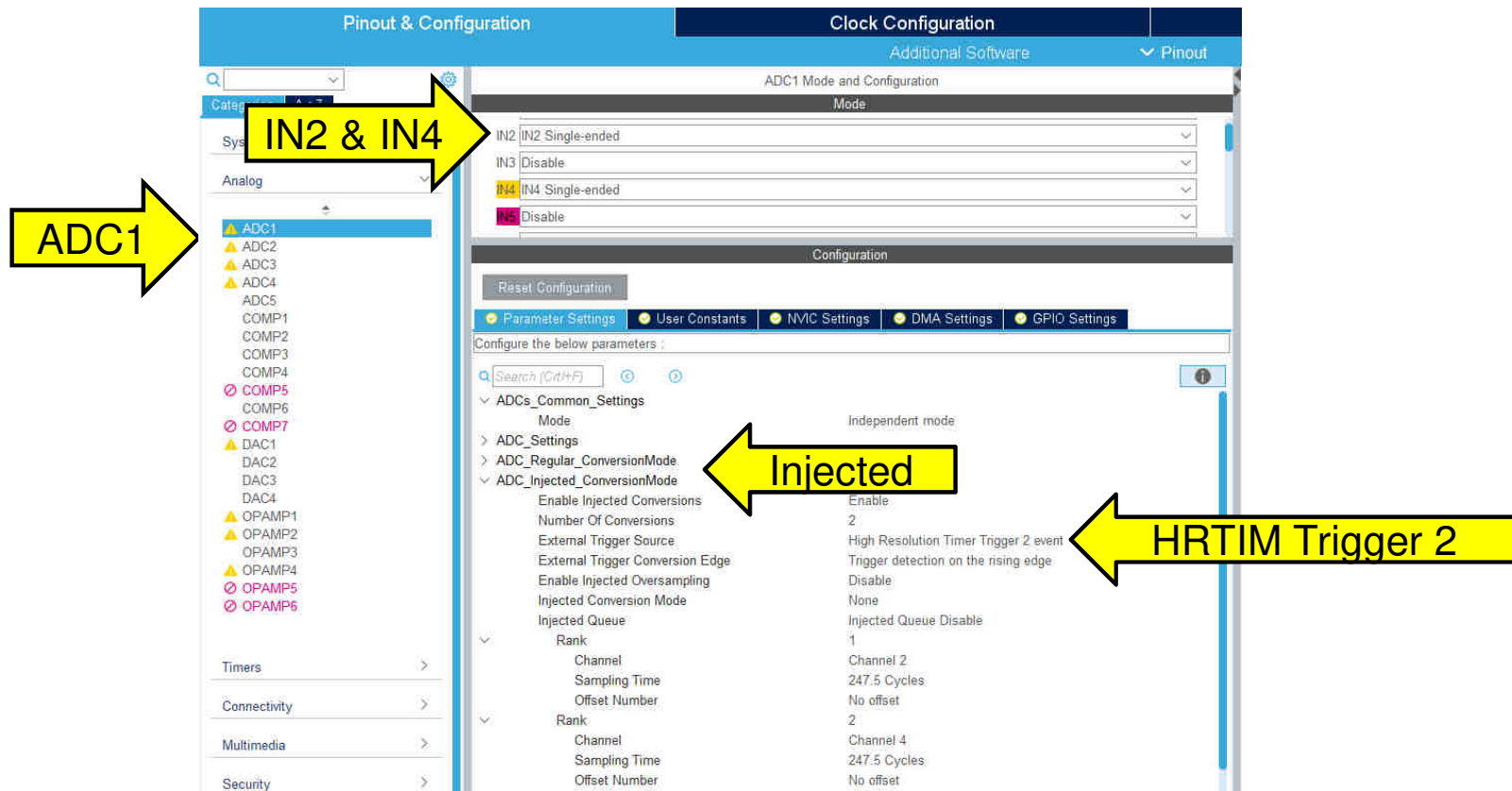
Configuration

Reset Configuration

DMA Settings	GPIO Settings
User Constants	NVIC Settings
Burst Mode Configuration	Timer C
Fault Lines Configuration	Timer D
High Resolution	ADC Triggers Configuration
HRTIM Interrupt Configuration	External Event Configuration
	Synchro Configuration

Configure the below parameters :

HRTIM – ADC – 2 injected input channels



ADC1

IN2 & IN4

Injected

HRTIM Trigger 2

Pinout & Configuration

Clock Configuration

Additional Software

ADC1 Mode and Configuration

Mode

IN2 IN2 Single-ended

IN3 Disable

IN4 IN4 Single-ended

IN5 Disable

Configuration

Reset Configuration

Parameter Settings

User Constants

NVIC Settings

DMA Settings

GPIO Settings

Configure the below parameters :

Search (Ctrl+F)

ADCs_Common_Settings

Mode

Independent mode

ADC_Settings

ADC_Regular_ConversionMode

ADC_Injected_ConversionMode

Enable Injected Conversions

Number Of Conversions

External Trigger Source

External Trigger Conversion Edge

Enable Injected Oversampling

Injected Conversion Mode

Injected Queue

Rank

Channel

Sampling Time

Offset Number

Rank

Channel

Sampling Time

Offset Number

Enable

2

High Resolution Timer Trigger 2 event

Trigger detection on the rising edge

Disable

None

Injected Queue Disable

1

Channel 2

247.5 Cycles

No offset

2

Channel 4

247.5 Cycles

No offset



Project Manager – Keil MDK-ARM V5

STM32CubeMX HRTIM_Buck_Boost.ioc: STM32G474REt: NUCLEO-G474RE

File Window Help

Home STM32G474REt: - NUCLEO-G474RE HRTIM_Buck_Boost.ioc - Project Manager GENERATE CODE

Pinout & Configuration Clock Configuration Project Manager Tools

Project

Project Settings

Project Name: HRTIM_Buck_Boost

Project Location: C:\Users\jeffery.blauser\STM32Cube\Repository\STM32Cube_FW_G4_V1.1.0\Projects\B-G474E-DPOW1\Examples_MIX\HRTIM_Buck_Boost\

Application Structure: Basic ☐ Do not generate the main()

Code Generator

Toolchain Folder Location: C:\Users\jeffery.blauser\STM32Cube\Repository\STM32Cube_FW_G4_V1.1.0\Projects\B-G474E-DPOW1\Examples_MIX\HRTIM_Buck_Boost\

Toolchain / IDE: MDK-ARM V5

Advanced Settings

Linker Settings

Minimum Heap Size: 0x200

Minimum Stack Size: 0x400

Mcu and Firmware Package

Mcu Reference: STM32G474REt

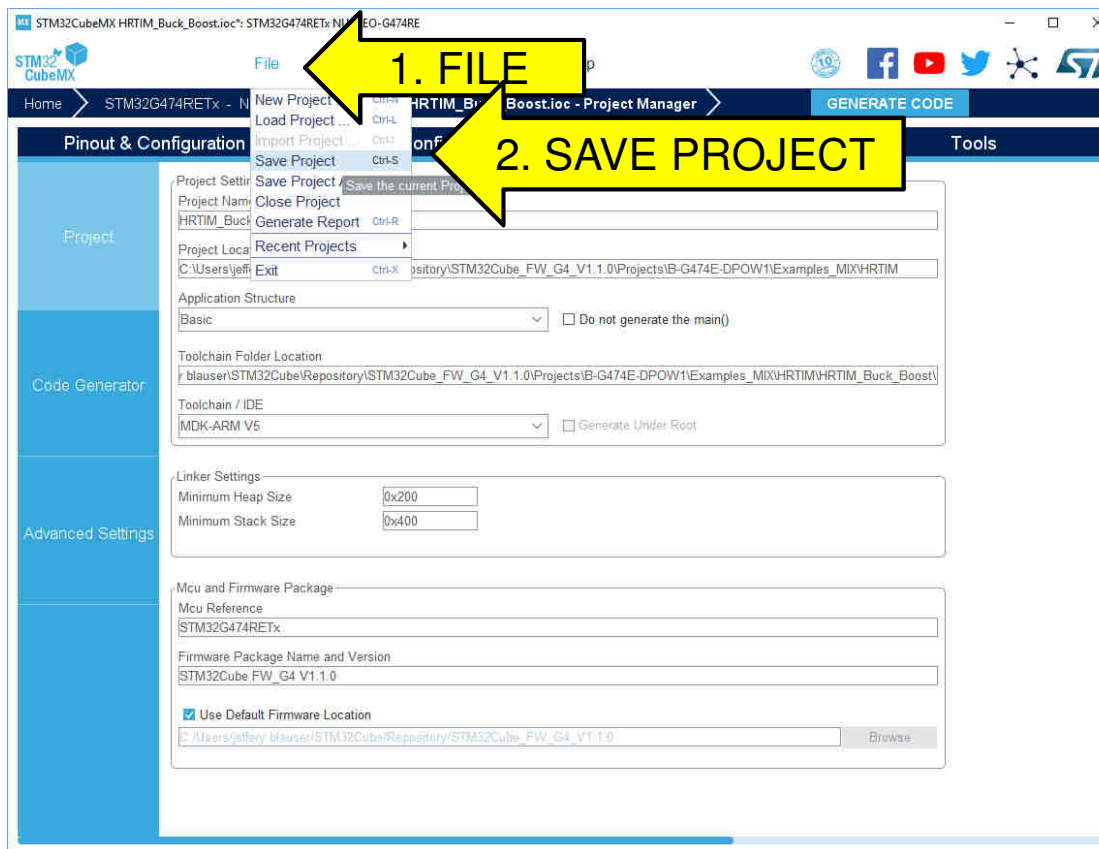
Firmware Package Name and Version: STM32Cube_FW_G4_V1.1.0

☒ Use Default Firmware Location

C:\Users\jeffery.blauser\STM32Cube\Repository\STM32Cube_FW_G4_V1.1.0 Browse

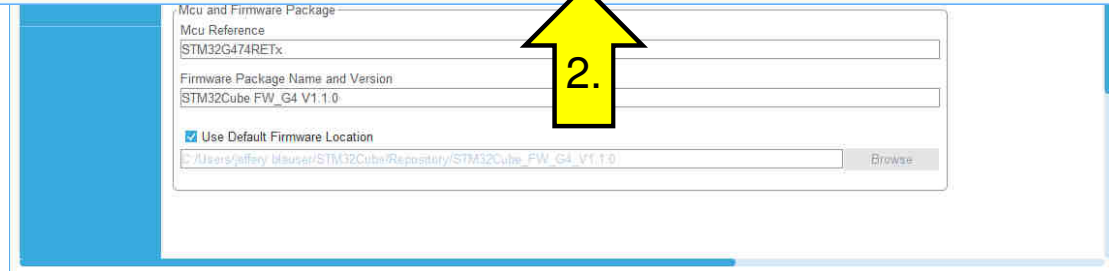
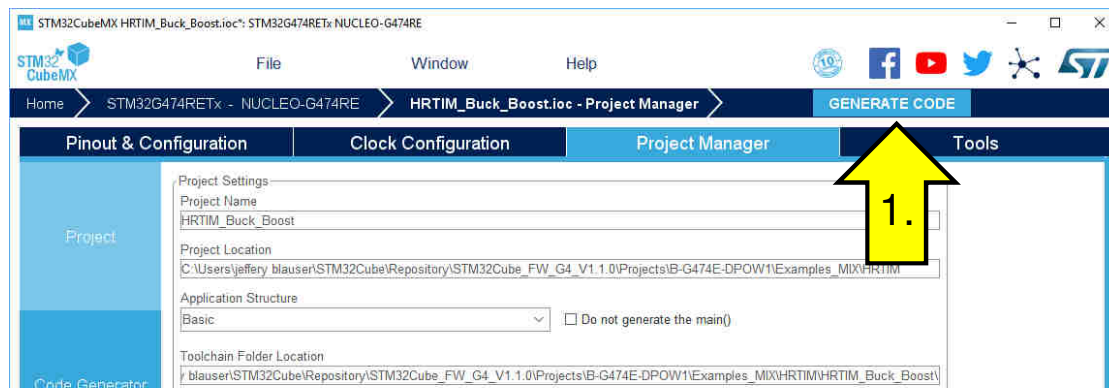


FILE→SAVE PROJECT





Generate Code





Keil Start up page

```
51 /* BUCK cannot re-start until Vout capacitor is de-energized below this value */
52 #define DE_ENERGIZING_THRESHOLD ((uint16_t)2500)
53
54 /* BOOST cannot be started if VIN value is above this threshold */
55 #define OVER_VOLTAGE_PROTECTION ((uint16_t)5000)
56
57 /* This is the VDDA supply in mV */
58 #define VDDA ((uint16_t)3300)
59
60 /* Deadtime values for buck and boost stages */
61 #define DT_RISING ((uint16_t)250) // 184ns
62 #define DT_FALLING ((uint16_t)300) // 220ns
63
64 __IO uint32_t Vin, Vout; /* volatile is here for debug convenience */
65
66 uint16_t Vin_Display; //Vin_Display made global to appear in the Logic Analyzer
67 uint16_t Vout_Display; //Vout_Display made global to appear in the Logic Analyzer
68
69 DemoState_TypeDef DemoState;
70
71 /* USER CODE END PV */
72
73 /* Private function prototypes -----*/
74 void SystemClock_Config(void);
75 static void MX_GPIO_Init(void);
76 static void MX_HRTIM1_Init(void);
77 static void MX_ADC1_Init(void);
78 /* USER CODE BEGIN PFP */
79
80 /* USER CODE END PFP */
81
82 /* Private user code -----*/
83 /* USER CODE BEGIN 0 */
84
85 /* USER CODE END 0 */
86
```

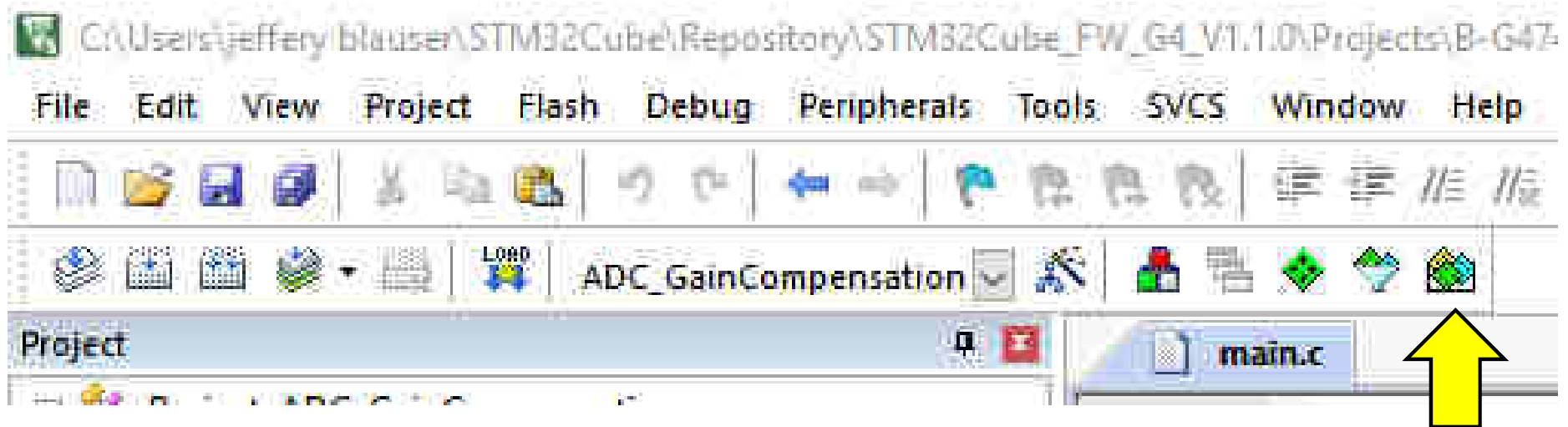




Keil MDK-ARM

226

Check the Device Pack





G4 Device Pack

The screenshot shows the Keil MDK-ARM Pack Installer window. The left pane displays a tree view of devices, with the STM32G474 device selected under the STM32G4 Series. The right pane shows a list of packs, with the Keil:STM32G4xx_DFP pack highlighted. A yellow arrow points to the 'Exit' button in the top right corner. Another yellow arrow points to the 'Keil:STM32G4xx_DFP' pack in the right pane. A third yellow arrow points to the 'STM32G474' device in the left pane. The bottom pane shows the output log with error messages related to downloading pack descriptions.

Device: STM32G474

Search: / Summary

Device

- Silicon Labs: 906 Devices
- Sinowealth: 1 Device
- SONIX: 50 Devices
- STMicroelectronics: 1284 Devices
 - STBlueNRG Series: 1 Device
 - STBlueNRG-1 Series: 1 Device
 - STBlueNRG-2 Series: 1 Device
 - STM32F0 Series: 111 Devices
 - STM32F1 Series: 95 Devices
 - STM32F2 Series: 46 Devices
 - STM32F3 Series: 90 Devices
 - STM32F4 Series: 205 Devices
 - STM32F7 Series: 115 Devices
 - STM32G0 Series: 34 Devices
 - STM32G4 Series: 118 Devices
 - STM32G431: 24 Devices
 - STM32G441: 8 Devices
 - STM32G471: 17 Devices
 - STM32G473: 25 Devices
 - STM32G474: 25 Devices
 - STM32G483: 9 Devices
 - STM32G484: 9 Devices
 - STM32GBK1: 1 Device
 - STM32H7 Series: 22 Devices
 - STM32L0 Series: 152 Devices
 - STM32L1 Series: 81 Devices

Packs

Pack	Action	Description
Device Specific	1 Pack	STM32G4xx_DFP
Generic	40 Packs	
Alibaba:AliOSThings	Install	AliOS Things software pack
ARM:AMP	Install	Software components for inter processor communication (Asymmetric Multi
ARM:CMSIS	Up to date	CMSIS (Cortex Microcontroller Software Interface Standard)
ARM:CMSIS-Driver	Up to date	CMSIS Drivers for external devices
ARM:CMSIS-Driver_Val...	Up to date	CMSIS-Driver Validation
ARM:CMSIS-FreeRTOS	Up to date	Bundle of FreeRTOS for Cortex-M and Cortex-A
ARM:CMSIS-RTOS_Val...	Up to date	CMSIS-RTOS Validation
ARM:mbedClient	Install	ARM mbed Client for Cortex-M devices
ARM:mbedTLS	Install	ARM mbed Cryptographic and SSL/TLS library for Cortex-M devices
ARM:minar	Install	mbed OS Scheduler for Cortex-M devices
ARM:TFM	Install	Trusted Firmware-M (TF-M) is the reference implementation of Arm's Platfor
EmbeddedOffice:Flexi...	Install	Flexible Safety RTOS
Keil:ARM_Compiler	Up to date	Keil ARM Compiler extensions for ARM Compiler 5 and ARM Compiler 6
Keil:IMXRT105x_MWP	Install+	NXP i.MX RT 105x MDK-Middleware examples and CMSIS-Drivers
Keil:Jansson	Install	Jansson is a C library for encoding, decoding and manipulating JSON data
Keil:MDK-Middleware	Up to date	Middleware for Keil MDK-Professional and MDK-Plus
lwIP:lwIP	Install	lwIP is a light-weight implementation of the TCP/IP protocol suite
MDK-Packs:AWS_IoT_...	Install+	SDK for connecting to AWS IoT from a device using embedded C
MDK-Packs:Azure_IoT	Install+	Microsoft Azure IoT SDKs and Libraries
MDK-Packs:cJSON	Install	Ultralightweight JSON parser in ANSI C
MDK-Packs:Google_Io...	Install+	Google Cloud IoT Device Connector
MDK-Packs:IoT_Socket	Install	Simple IP Socket (BSD like)
MDK-Packs:Jsmin	Install	Minimalistic JSON parser/tokenizer in C
MDK-Pack:PaHo-MQ...	Install+	Embedded MQTT C/C++ Client Libraries

Output

Cannot download file http://mcuxpresso.nxp.com/cmsis_pack/repo/NXP.LPC54018M_DFP.pdsc: Illegal MIME type 'text/html'

Cannot download file http://mcuxpresso.nxp.com/cmsis_pack/repo/NXP.LPC54018_DFP.pdsc: Illegal MIME type 'text/html'

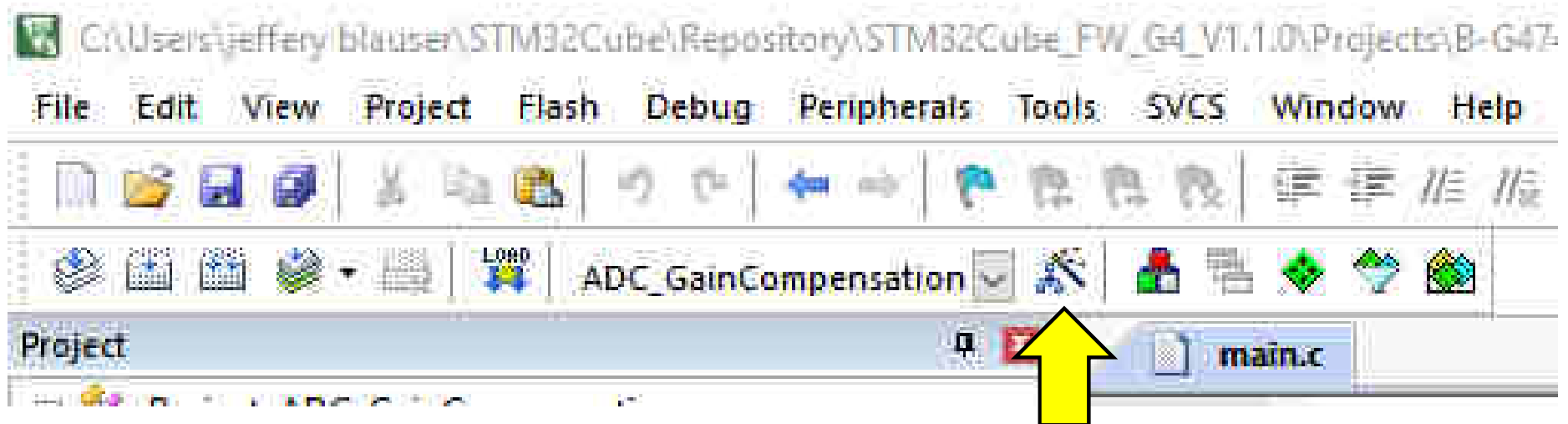
Cannot download file http://mcuxpresso.nxp.com/cmsis_pack/repo/NXP.LPC54101_DFP.pdsc: Illegal MIME type 'text/html'

Action [2 left]: Update Pack descriptions, download http://mcuxpresso.nxp.com/cmsis_pack/repo/NXP.LPC54102_DFP.pdsc

0% ONLINE

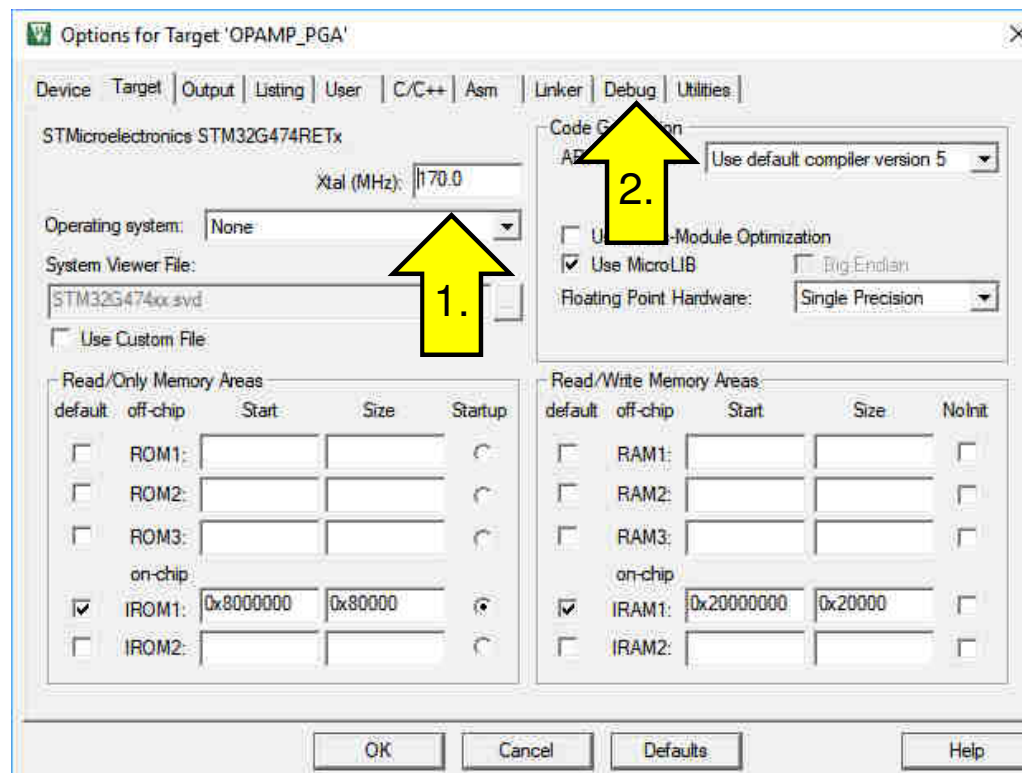


Check the Options



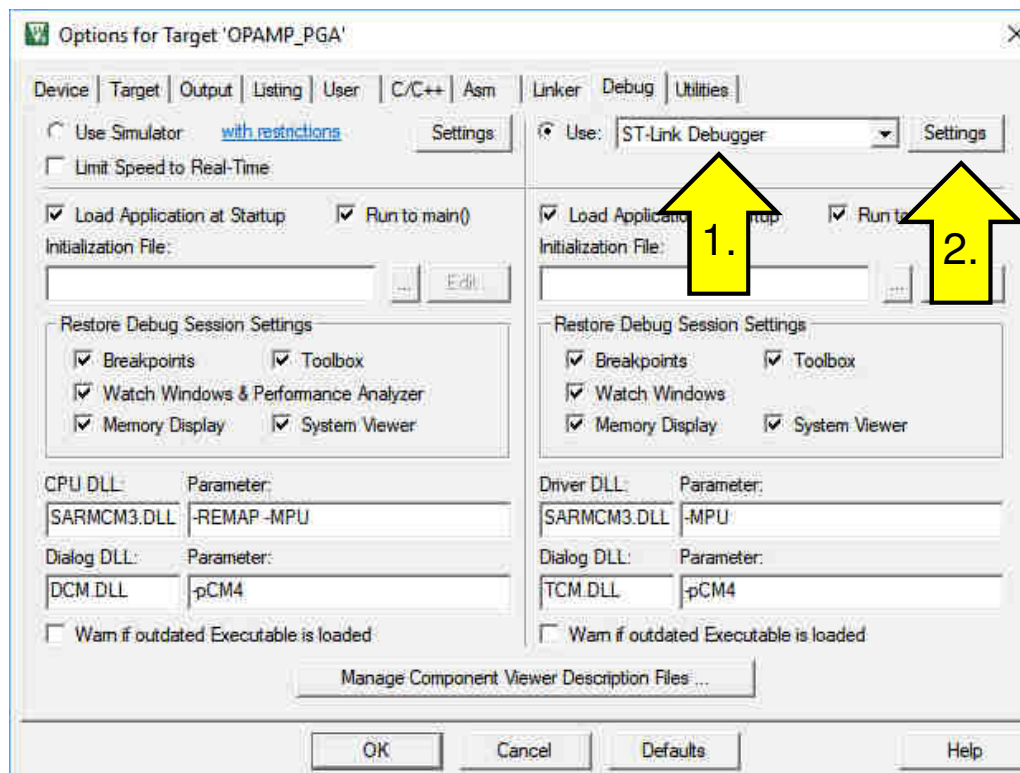


Clock speed and Debug



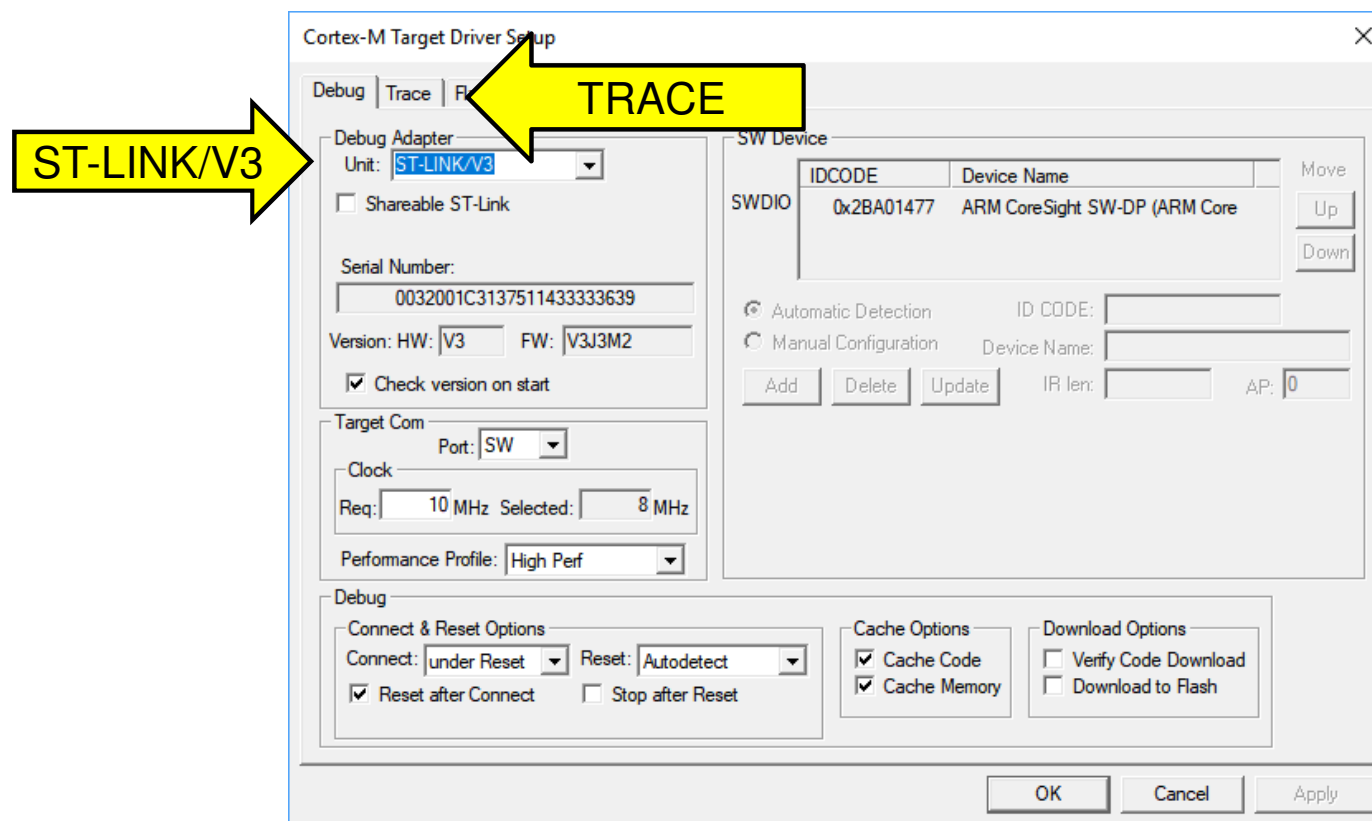


ST-Link Debugger - Settings





ST-Link/V3 & TRACE





Core Clock & TRACE Enable (After)

The screenshot shows the 'Cortex-M Target Driver Setup' dialog box with the 'Trace' tab selected. The following annotations highlight specific settings:

- 1. 170**: Points to the 'Core Clock' field, which is set to 170.000000 MHz.
- 2. TRACE Enable**: Points to the 'Trace Enable' checkbox, which is checked.
- 3. Uncheck EXCTRC**: Points to the 'EXCTRC: Exception Tracing' checkbox in the 'Trace Events' section, which is unchecked.
- 4. Uncheck all but 15**: Points to the 'ITM Stimulus Ports' section, where only port 15 is checked in the 'Enable' row.
- 5. OK**: Points to the 'OK' button at the bottom right of the dialog.

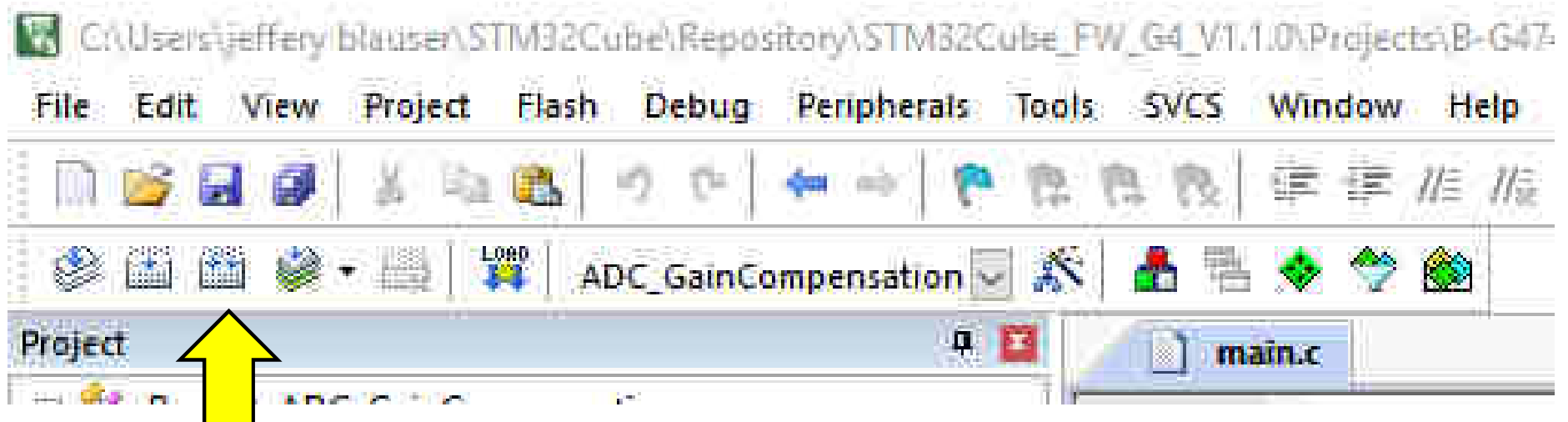
The dialog box contains the following sections and settings:

- Debug** | **Trace** | Flash Download | Pack
- Core Clock**: 170.000000 MHz
- Trace Port**: Serial Wire Output - UART/NRZ
- SWO Clock Prescaler**: 15
- Autodetect**: ☒
- SWO Clock**: 11.333333 MHz
- Timestamps**: ☒ Enable, Prescaler: 1
- PC Sampling**: Prescaler: 1024*16, ☐ Periodic, Period: <Disabled>, ☐ on Data R/W Sample
- Trace Events**:
 - ☐ CPI: Cycles per Instruction
 - ☐ EXC: Exception overhead
 - ☐ SLEEP: Sleep Cycles
 - ☐ LSU: Load Store Unit Cycles
 - ☐ FOLD: Folded Instructions
 - ☐ EXCTRC: Exception Tracing
- ITM Stimulus Ports**:

31	Port	24	23	Port	16	15	Port	8	7	Port	0
Enable: 0x00008000						☒					
Privilege: 0x00000008	Port 31..24	☒	Port 23..16	☐	Port 15..8	☐	Port 7..0	☐			
- Advanced settings**:
 - ☐ Ignore packets with no SYNC
 - ☐ Overwrite CYCCNT
- Buttons**: OK, Cancel, Apply



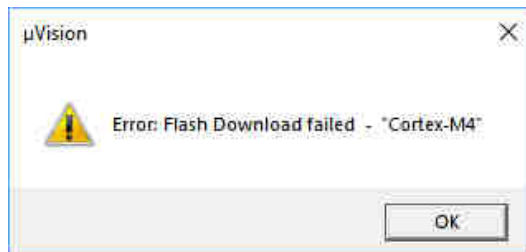
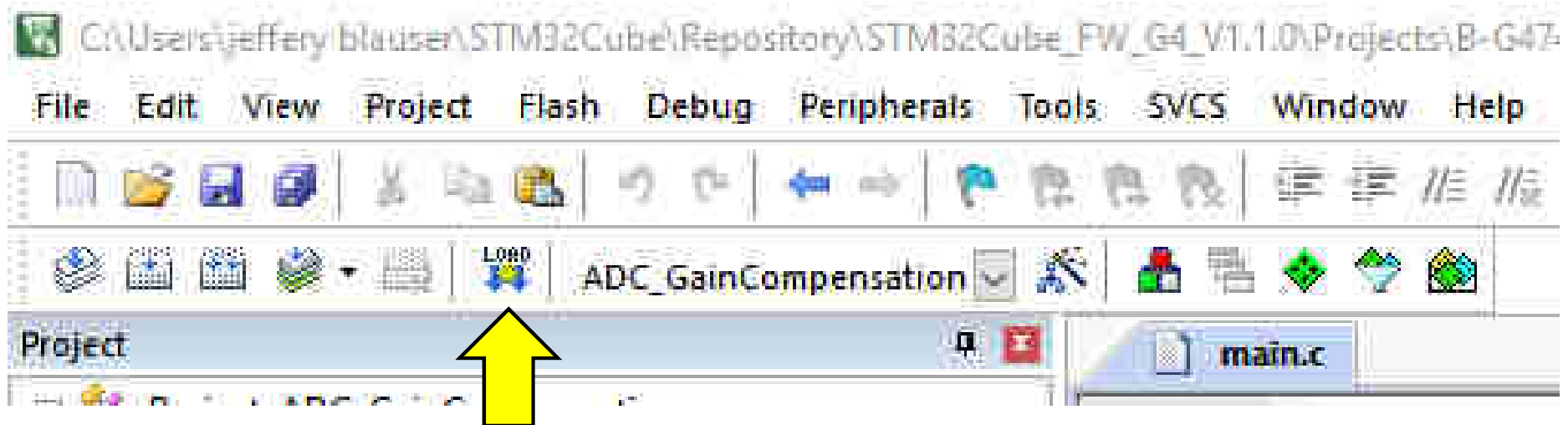
Build



Build Output

```
Program Size: Code=15028 RO-data=604 RW-data=92 ZI-data=1428  
FromELF: creating hex file...  
HRTIM_Buck_Boost\HRTIM_Buck_Boost.hex: Warning: Q9931W: Your license for feature Keil will expire in 29 days  
Finished: 0 information, 1 warning and 0 error messages.  
"HRTIM_Buck_Boost\HRTIM_Buck_Boost.axf" - 0 Error(s), 0 Warning(s).  
Build Time Elapsed: 00:01:33
```

Load



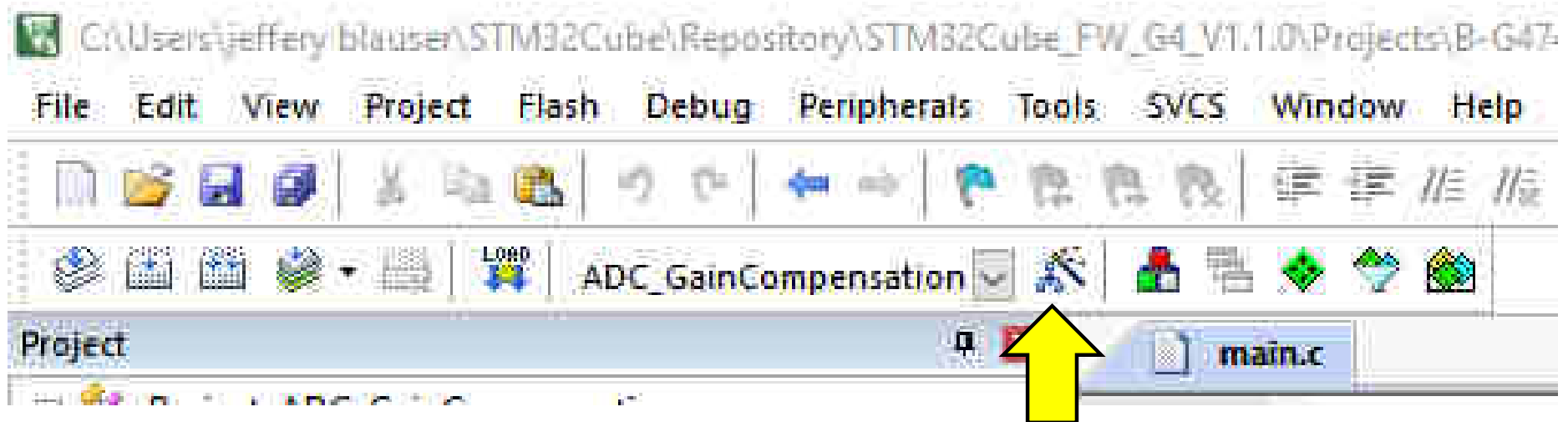
```
Build Output
Contents mismatch at: 08001860H (Flash=38H Required=C5H) !
Contents mismatch at: 08001861H (Flash=46H Required=F8H) !
Contents mismatch at: 08001862H (Flash=FEH Required=88H) !
Contents mismatch at: 08001863H (Flash=F7H Required=00H) !
Contents mismatch at: 08001864H (Flash=B9H Required=60H) !
Contents mismatch at: 08001865H (Flash=FCH Required=6CH) !
Too many errors to display !
Error: Flash Download failed - "Cortex-M4"
```



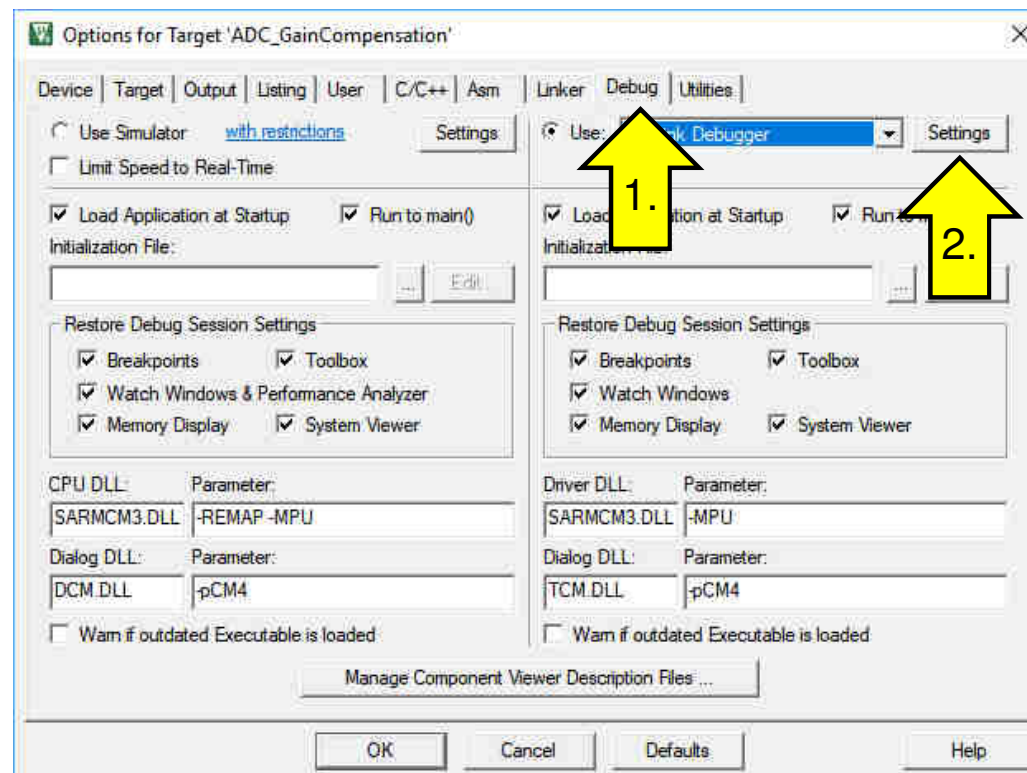
Keil MDK-ARM

235

Options

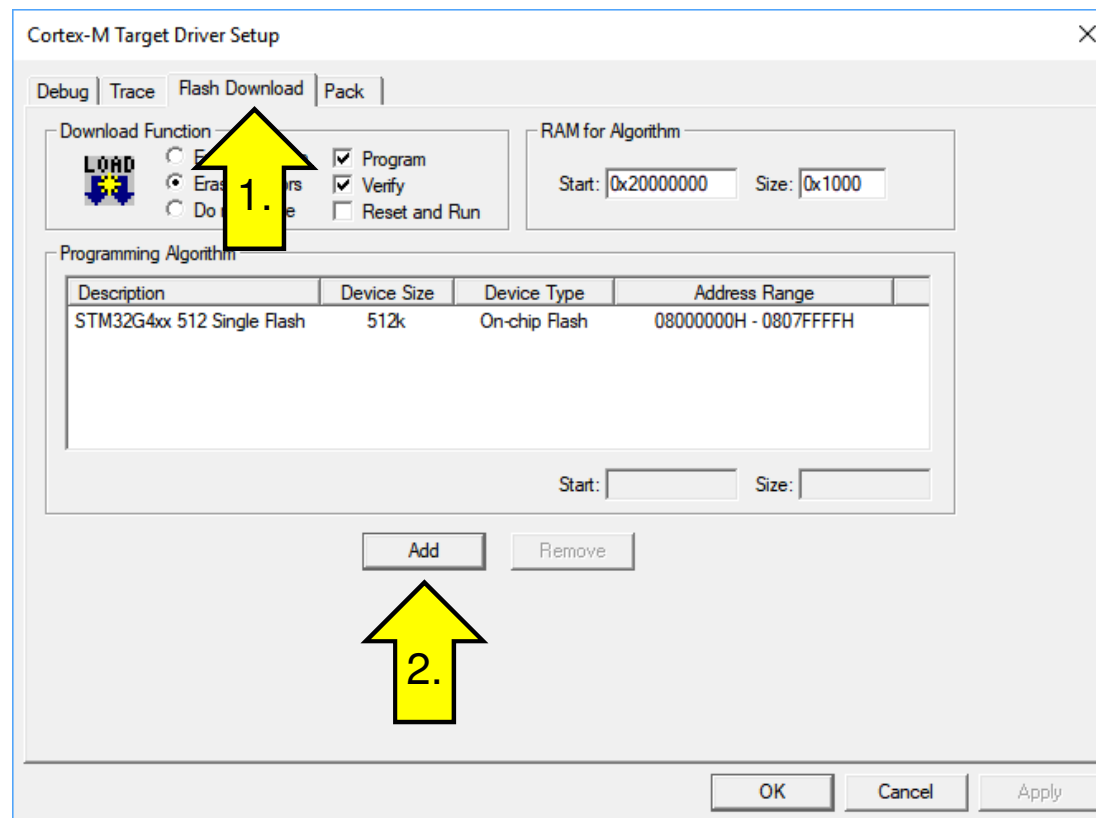


ST-Link Debugger - Settings



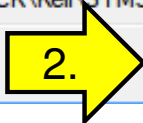
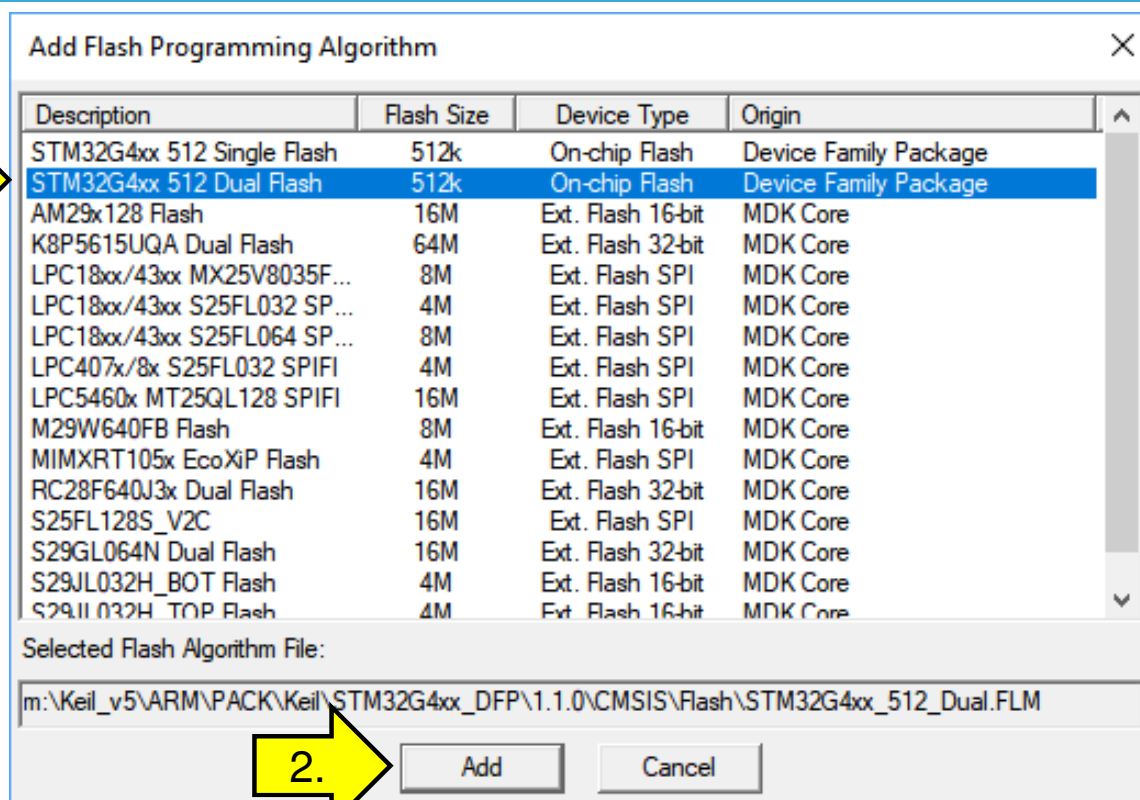


Flash Download → Add

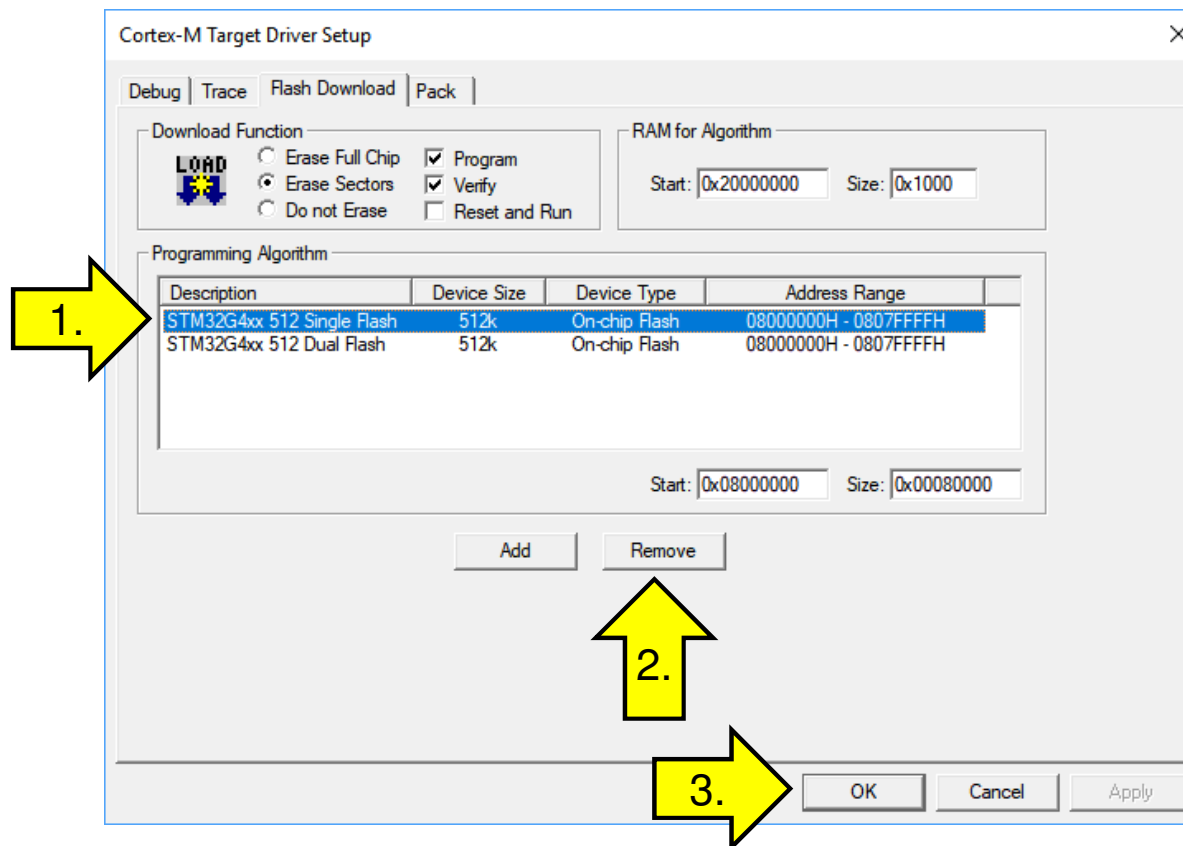




STM32G4xx 512 Dual Flash

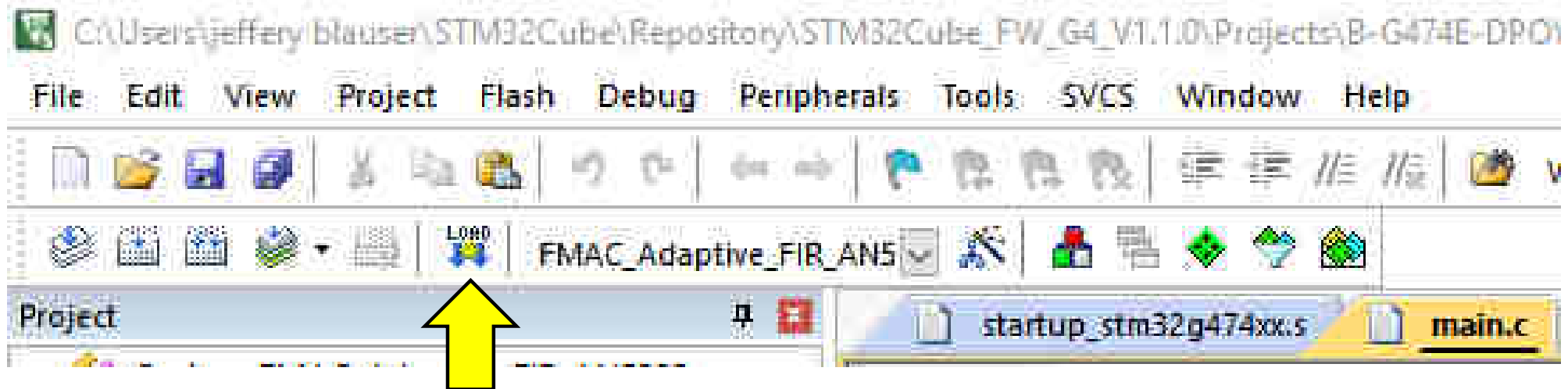


Remove Single Flash





Load



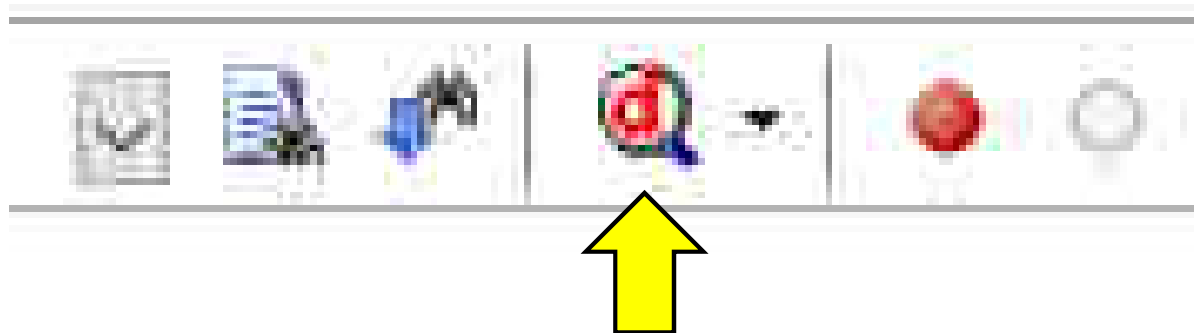
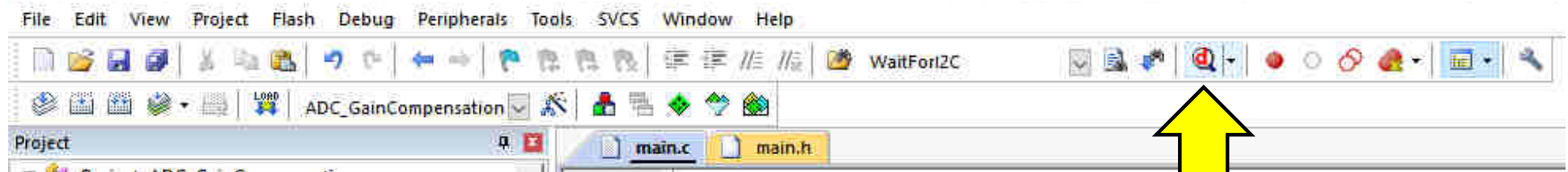
```
Build Output
Build Time Elapsed: 00:01:33
Load "HRTIM_Buck_Boost\\HRTIM_Buck_Boost.axf"
Erase Done.
Programming Done.
Verify OK.
Flash Load finished at 18:38:15
```




Keil MDK-ARM

241

Debugger

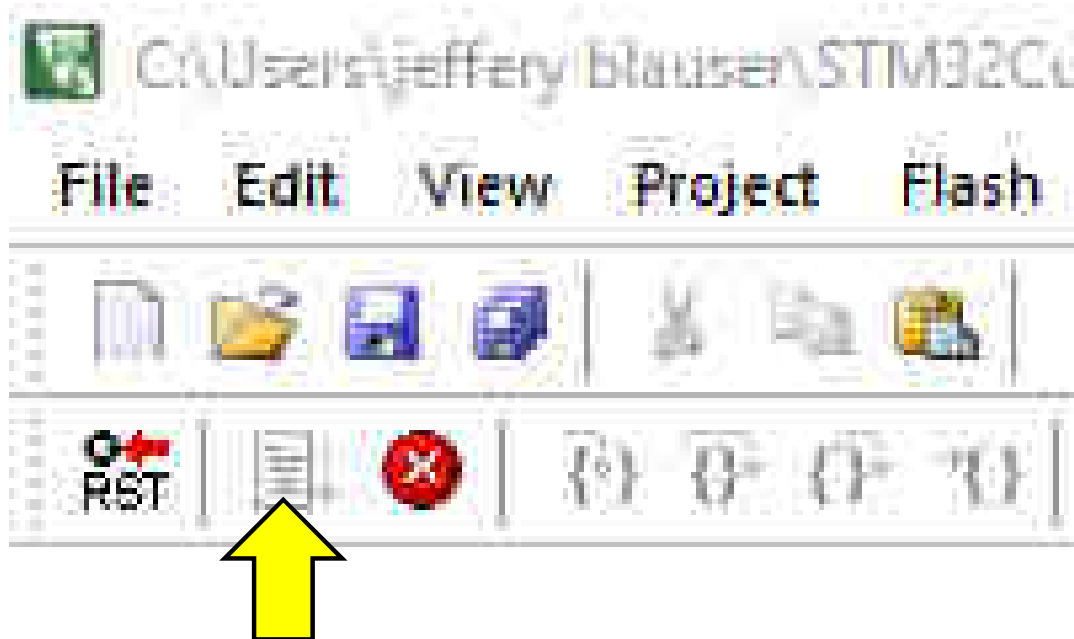




Keil MDK-ARM

242

RUN – Operating as a Buck converter.

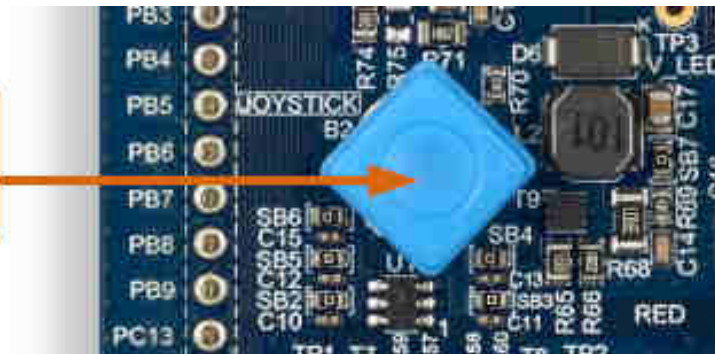


LD4 blinks rapidly



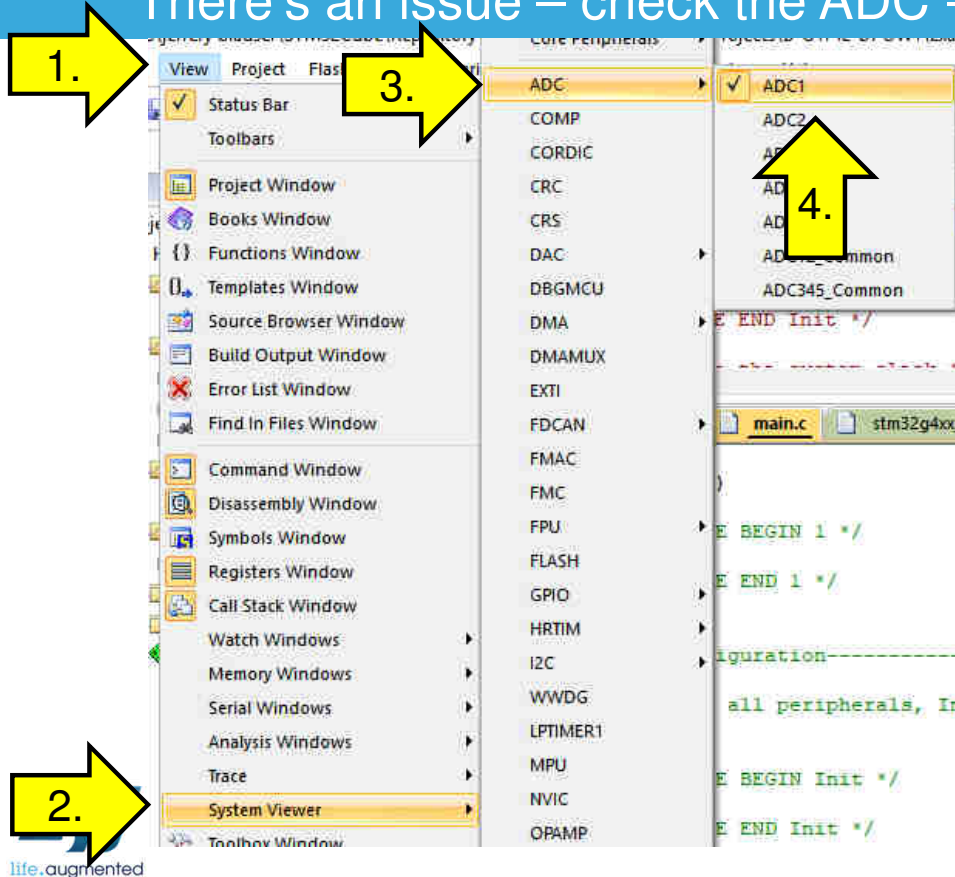
Press the Blue User Button for BOOST operation

B2
Joystick



LD3 blinks rapidly

There's an issue – check the ADC - VIEW→System Viewer→ADC→ADC1



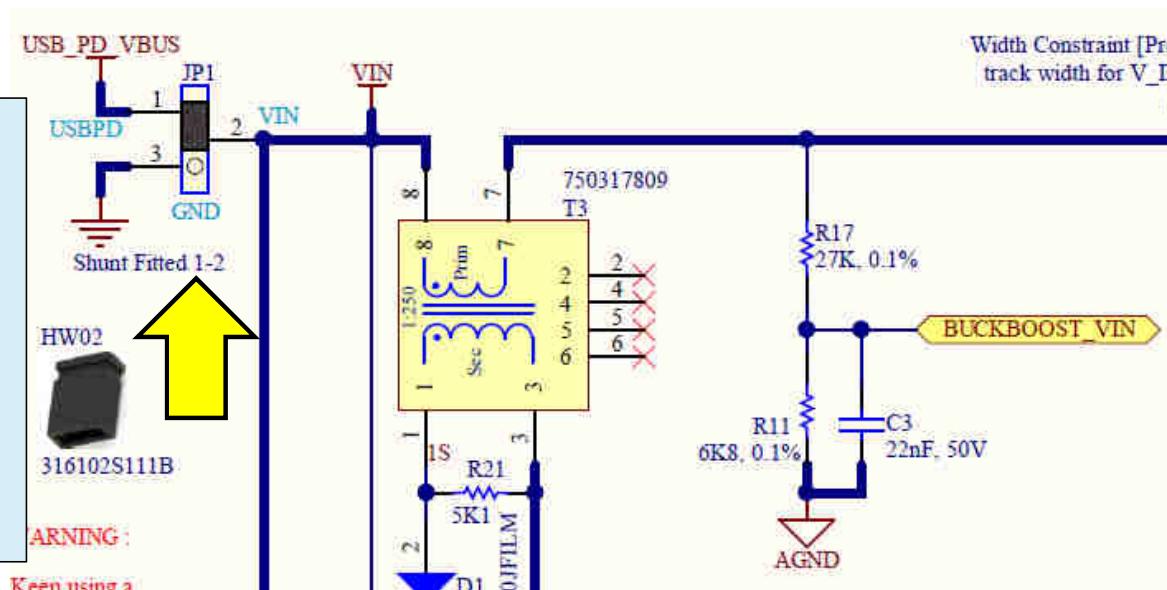
There are zeros in ADC1→JDR1 & 2 which are the two injected channel data registers

Uh Oh

ADC1	
Property	Value
ISR	0x00000061
IER	0
CR	0x10000009
CFGR	0x80000000
CFGR2	0
SMPR1	0x00006180
SMPR2	0
TR1	0x0FFF0000
TR2	0x00FF0000
TR3	0x00FF0000
SQR1	0x00000080
SQR2	0
SQR3	0
SQR4	0
DR	0
JSQR	0x000204CD
OFR1	0
OFR2	0
OFR3	0
OFR4	0
JDR1	0
JDR2	0
JDR3	0
JDR4	0

Schematic

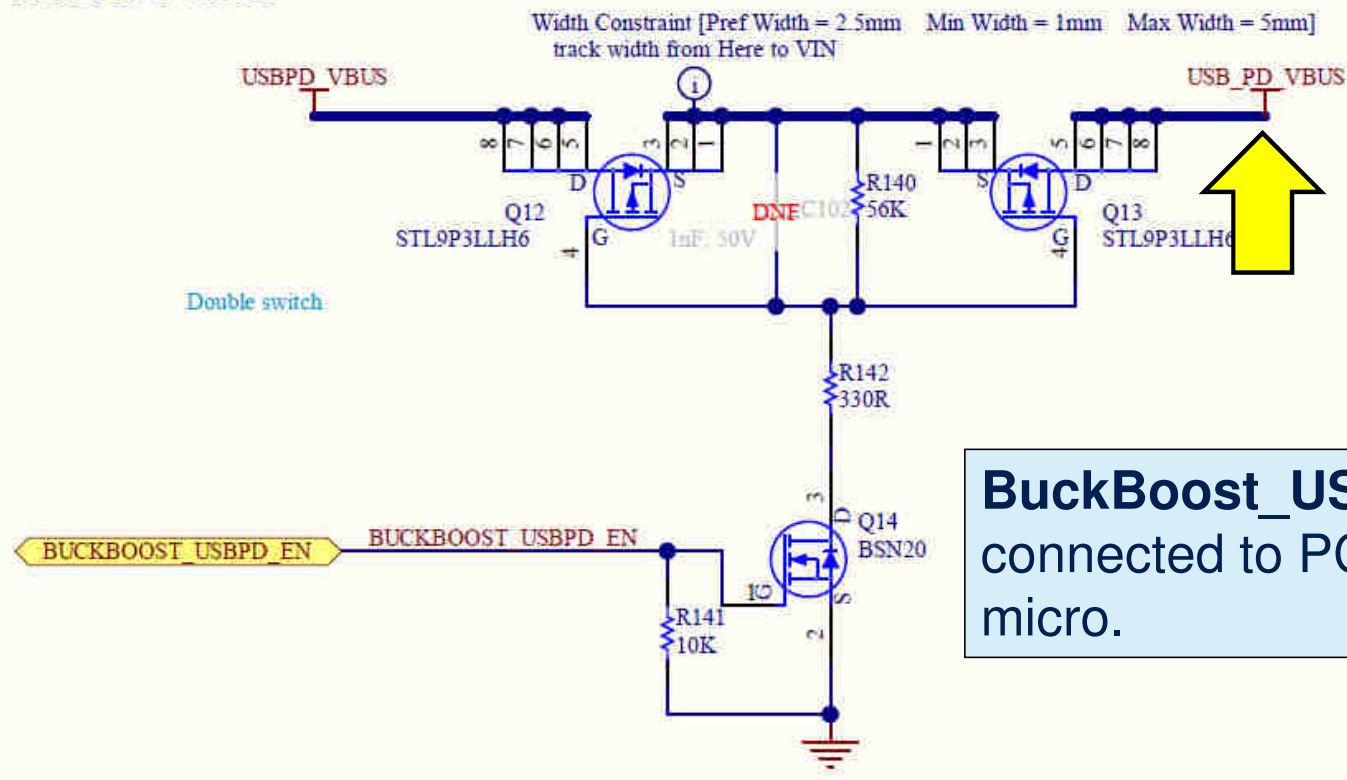
5V from USB-C is available here. Are JP1 pins 1 & 2 jumpered together? Keep **USB_PD_VBUS** in mind.



BuckBoost_VIN is connected to ADC1_IN2.

Schematic – Where do we find USB_PD_VBUS?

Double Switch





Check PC3 - VIEW→System Viewer→GPIO→GPIOC

1.

2.

3.

4.

Uh Oh

GPIOC output data registers do not seem to be configured so PC3 is not being driven and the MOSFETS are not enabled by BuckBoost_USBPD_EN

Property	Value
MODER	0xF3FFFFFF
OTYPER	0
OSPEEDR	0
PUPDR	0x04000000
IDR	0x00002000
ODR	0x00000000
ODR15	☐
ODR14	☐
ODR13	☐
ODR12	☐
ODR11	☐
ODR10	☐
ODR9	☐
ODR8	☐
ODR7	☐
ODR6	☐
ODR5	☐
ODR4	☐
ODR3	☐
ODR2	☐
ODR1	☐
ODR0	☐



Click on PC3→Choose GPIO_OutPut→System Core→GPIO

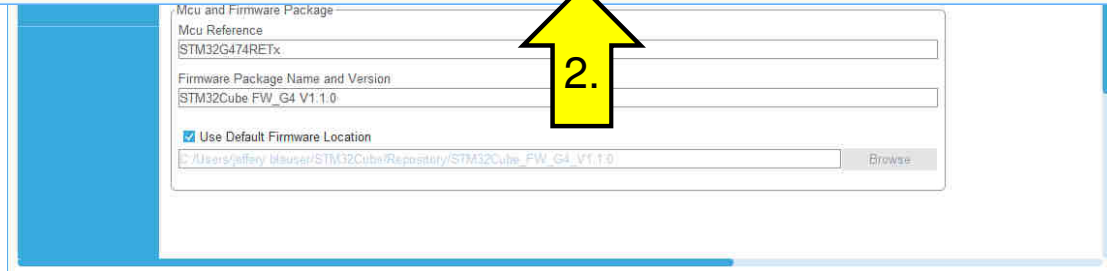
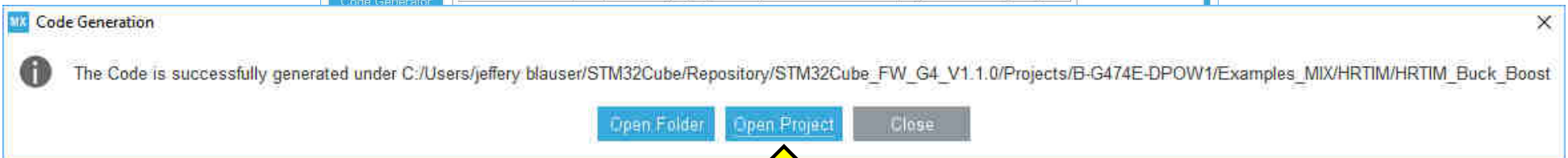
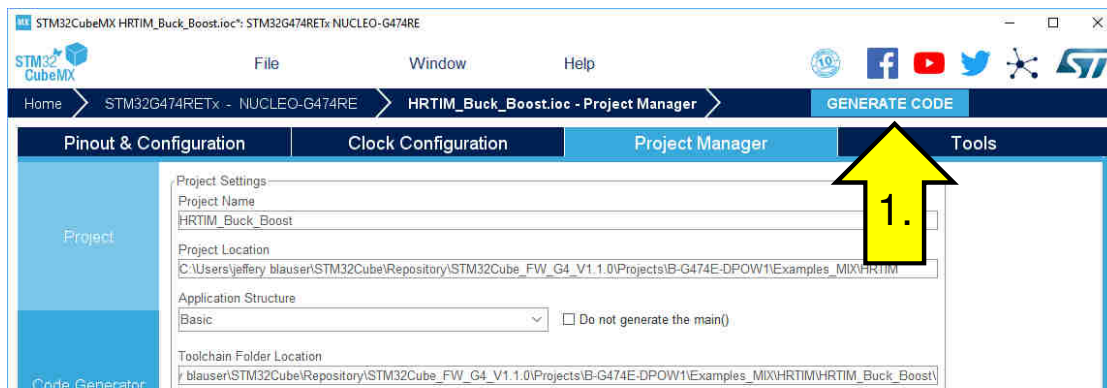
The screenshot shows the STM32CubeMX interface for the STM32G474REt6 NUCLEO-G474RE. The 'Pinout & Configuration' tab is active, displaying the 'GPIO Mode and Configuration' for PC3. The configuration is set to 'Output Push' with 'No pull-up or pull-down' and 'Very High' maximum output speed. The user label is 'BUCKBOOST_USBD_EN'. The pin is highlighted in the pinout diagram on the right.

Yellow arrows indicate the steps:

1. Click on PC3 in the pinout diagram.
2. Click on 'GPIO' in the 'System Core' list.
3. Click on 'GPIO' in the 'Additional Software' list.
4. Click on 'Output Push' in the 'GPIO mode' dropdown.
5. Click on 'No pull-up or pull-down' in the 'GPIO Pull-up/Pull-down' dropdown.



Generate Code



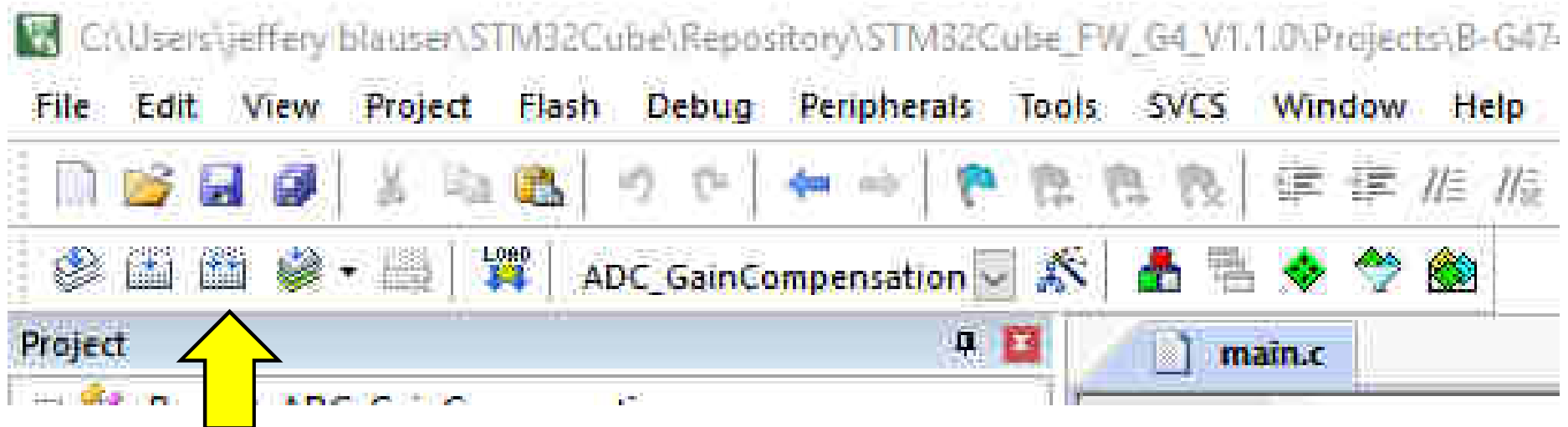


Main.c with new GPIOC code (you do not add this code)

```
601 {  
602     GPIO_InitTypeDef GPIO_InitStructure = {0};  
603  
604     /* GPIO Ports Clock Enable */  
605     __HAL_RCC_GPIOC_CLK_ENABLE();  
606     __HAL_RCC_GPIOA_CLK_ENABLE();  
607     __HAL_RCC_GPIOB_CLK_ENABLE();  
608  
609     /*Configure GPIO pin Output Level */  
610     HAL_GPIO_WritePin(BUCKBOOST_USBDP_EN_GPIO_Port, BUCKBOOST_USBDP_EN_Pin, GPIO_PIN_SET);  
611  
612     /*Configure GPIO pin Output Level */  
613     HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_RESET);  
614  
615     /*Configure GPIO pin : BUCKBOOST_USBDP_EN_Pin */  
616     GPIO_InitStructure.Pin = BUCKBOOST_USBDP_EN_Pin;  
617     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;  
618     GPIO_InitStructure.Pull = GPIO_NOPULL;  
619     GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_VERY_HIGH;  
620     HAL_GPIO_Init(BUCKBOOST_USBDP_EN_GPIO_Port, &GPIO_InitStructure);  
621  
622     /*Configure GPIO pin : LD2_Pin */  
623     GPIO_InitStructure.Pin = LD2_Pin;  
624     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;  
625     GPIO_InitStructure.Pull = GPIO_NOPULL;  
626     GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;  
627     HAL_GPIO_Init(LD2_GPIO_Port, &GPIO_InitStructure);  
628  
629 }
```



Build

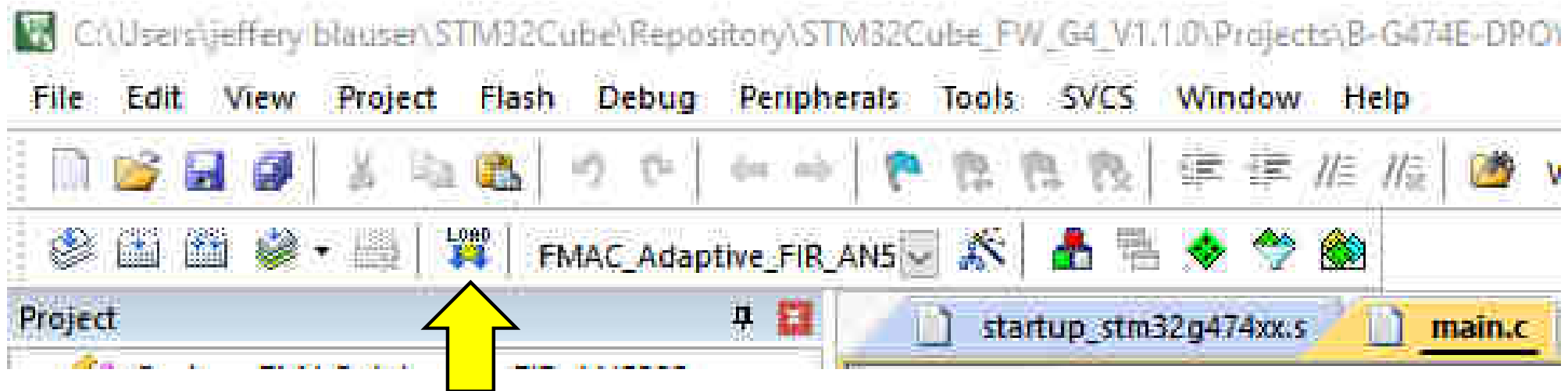


Build Output

```
Program Size: Code=15028 RO-data=604 RW-data=92 ZI-data=1428
FromELF: creating hex file...
HRTIM_Buck_Boost\HRTIM_Buck_Boost.hex: Warning: Q9931W: Your license for feature Keil will expire in 28 days
Finished: 0 information, 1 warning and 0 error messages.
"HRTIM_Buck_Boost\HRTIM_Buck_Boost.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:01:28
```



Load



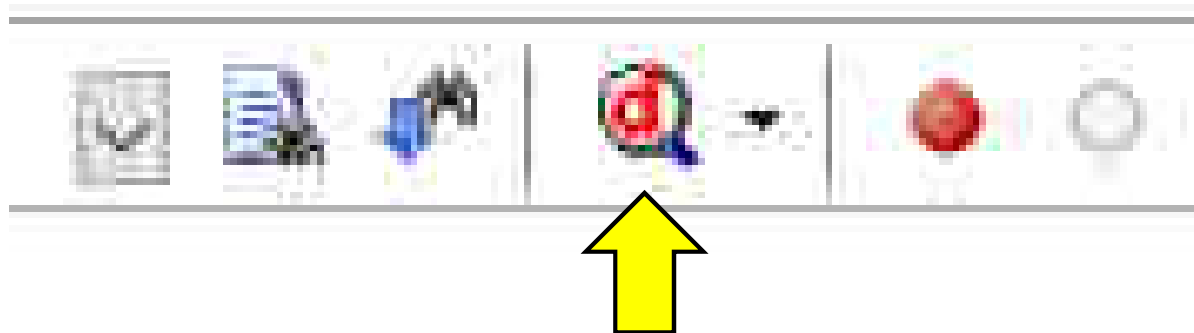
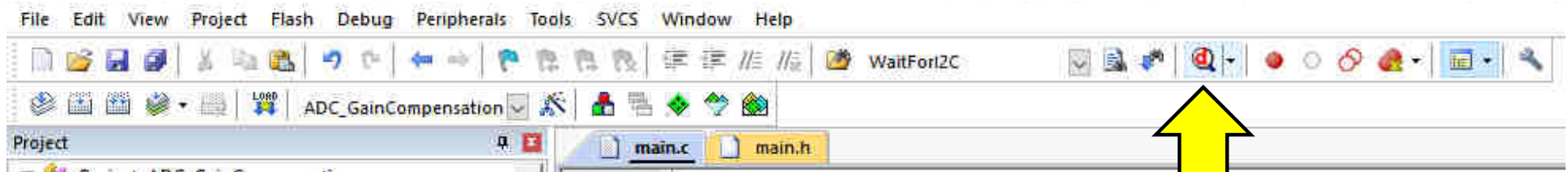
```
Build Output
Build Time Elapsed: 00:01:33
Load "HRTIM_Buck_Boost\\HRTIM_Buck_Boost.axf"
Erase Done.
Programming Done.
Verify OK.
Flash Load finished at 18:38:15
```



Keil MDK-ARM

253

Debugger

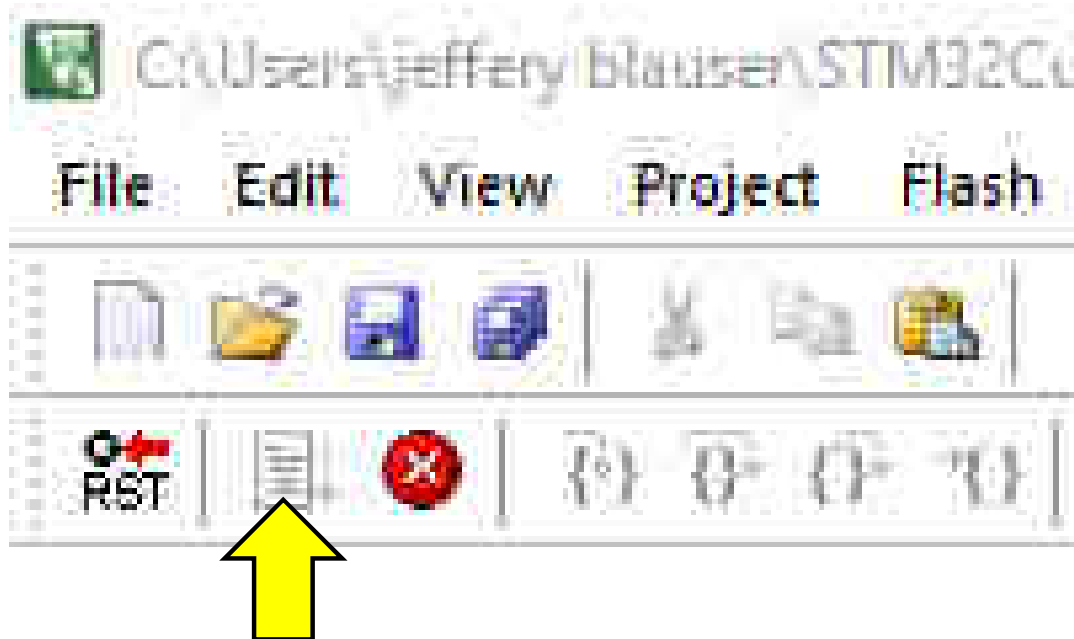




Keil MDK-ARM

254

RUN – Operating as a Buck converter.

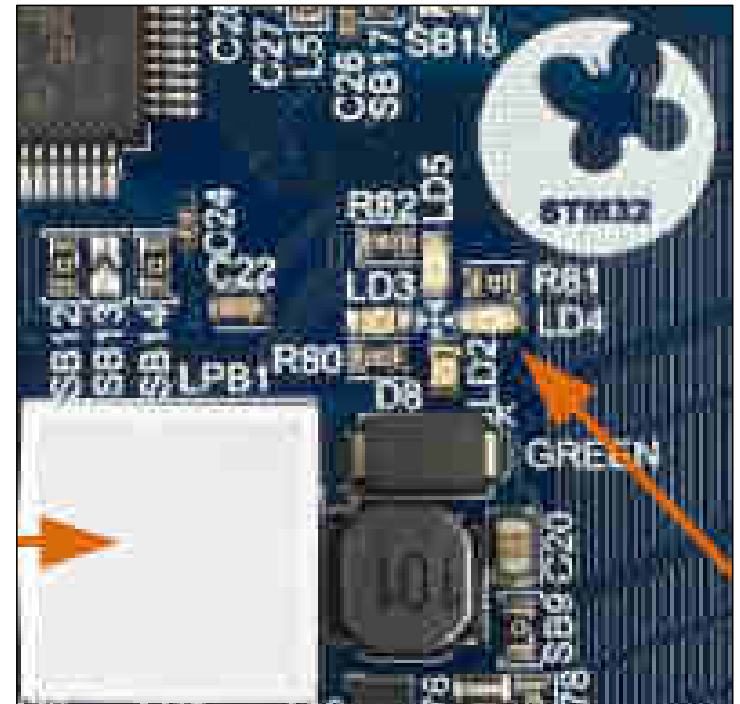
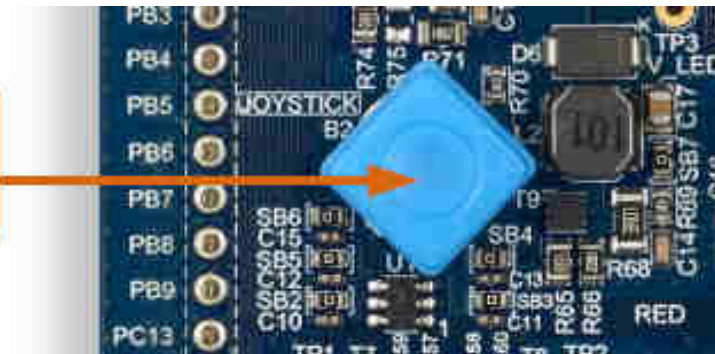


LD4 blinks rapidly



Press the Blue User Button for BOOST operation

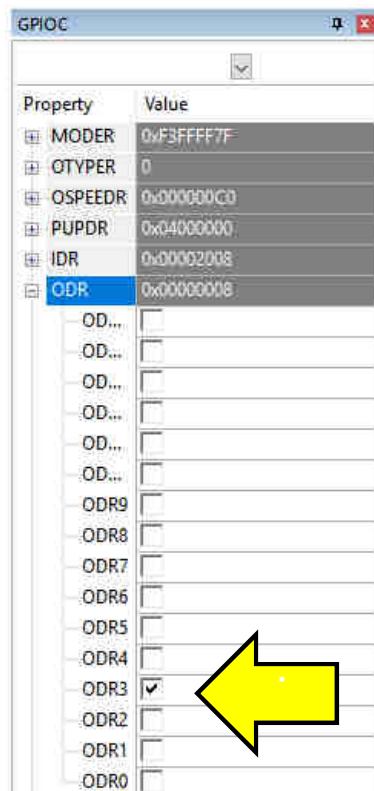
B2
Joystick



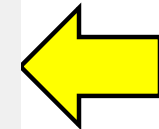
LD3 blinks rapidly



Check GPIOC – PC3 and ADC1 injected data registers

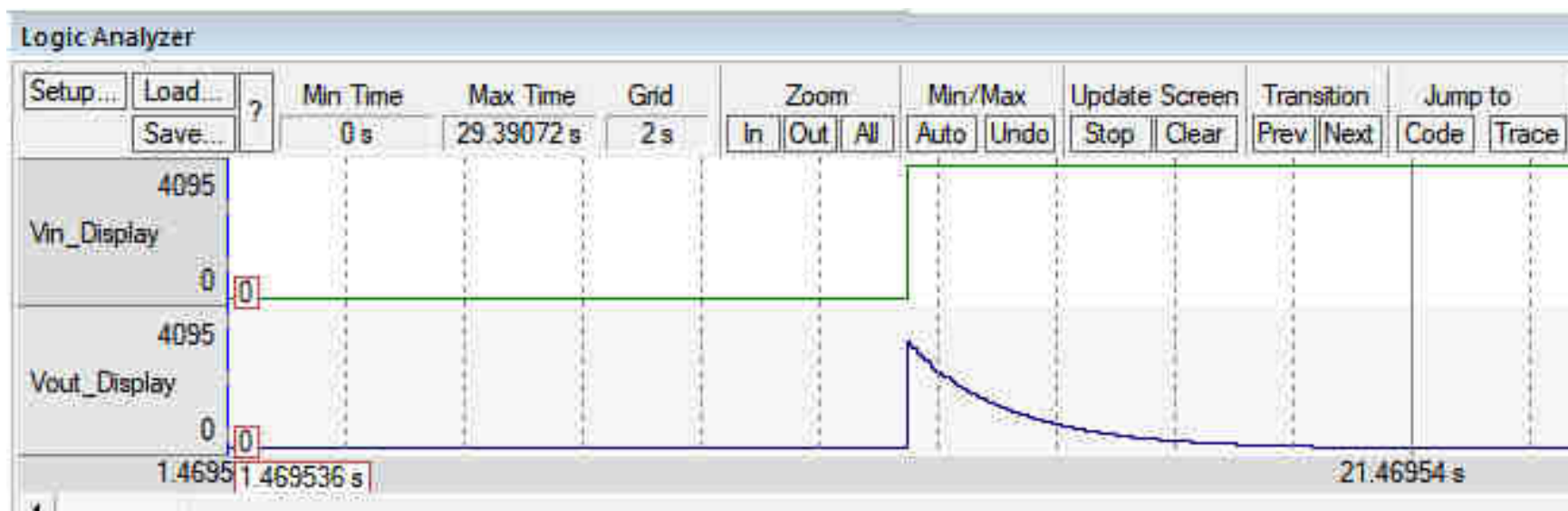


+ ODR4	0
+ JDR1	0x000004DD
+ JDR2	0x00000134
+ JDR3	0
+ JDR4	0



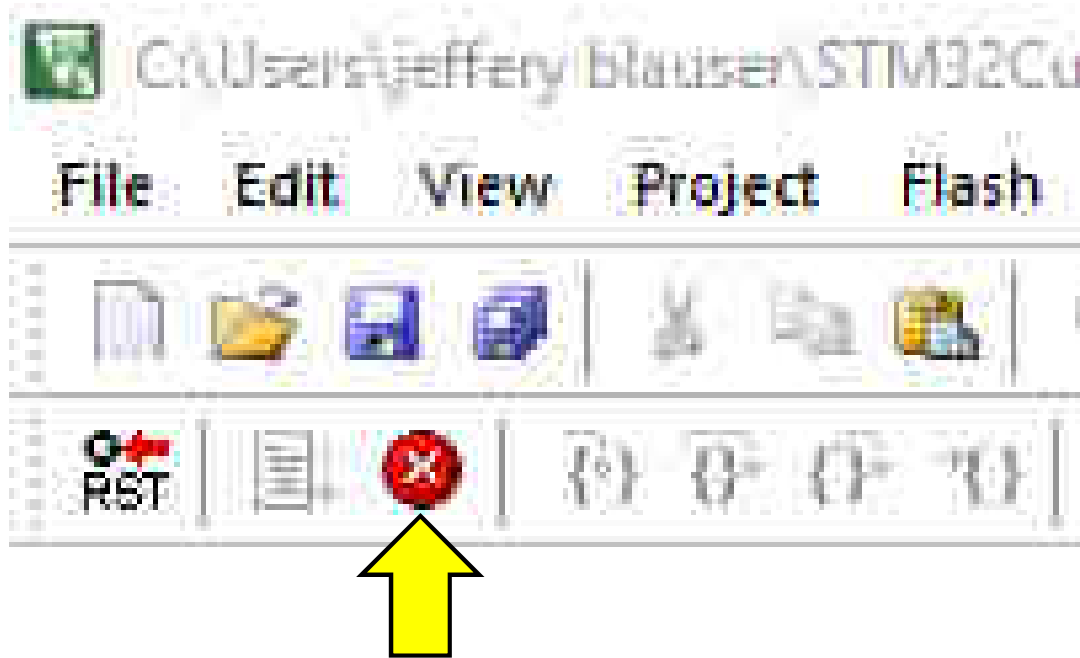


Logic Analyzer – Vin & Vout – Buck Mode



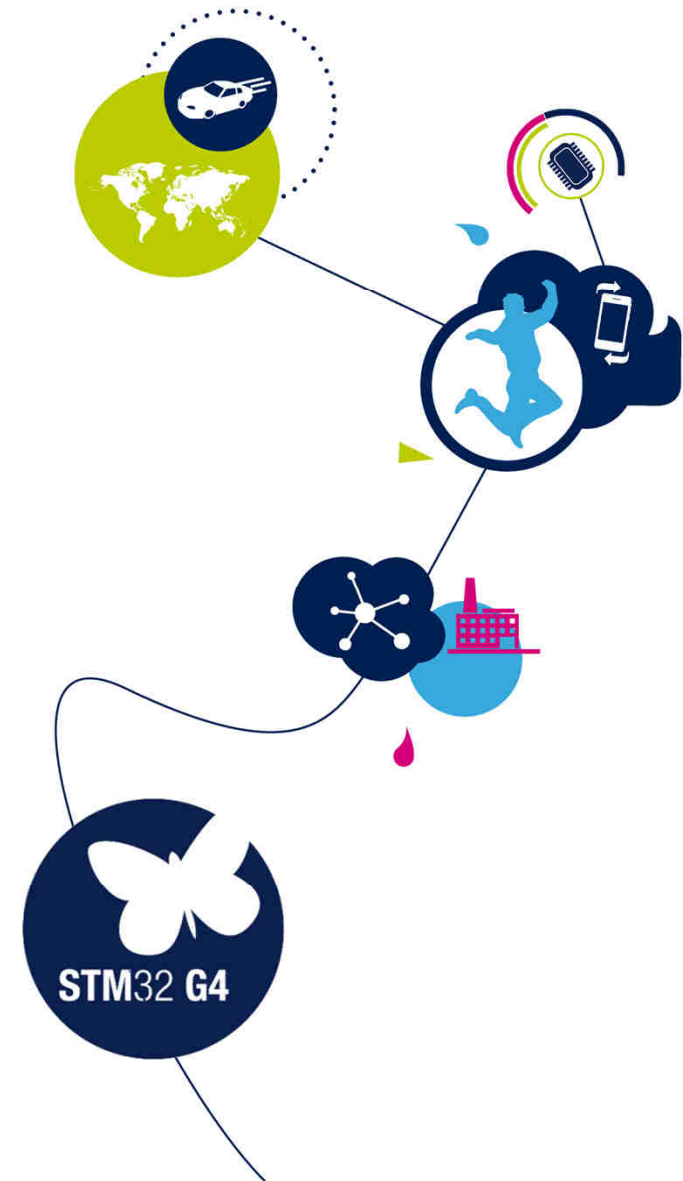


STOP Execution



STM32G4 mixed-signal MCU workshop

*Hands On:
CORDIC Hardware Accelerator*





CORDIC: Main features

260

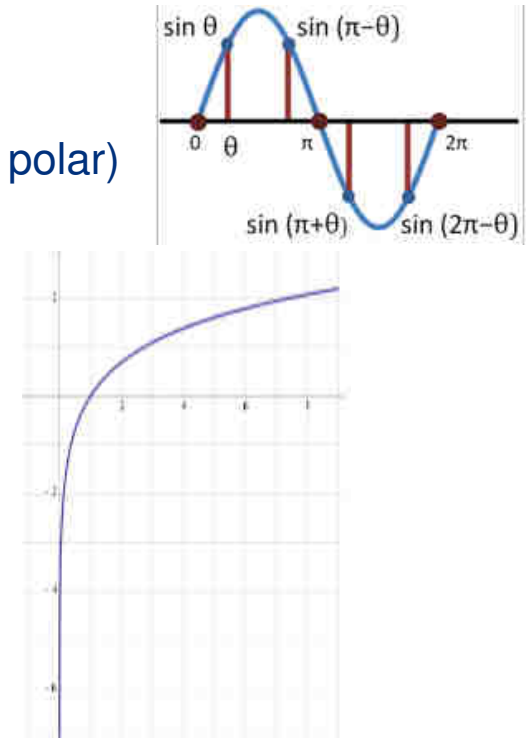
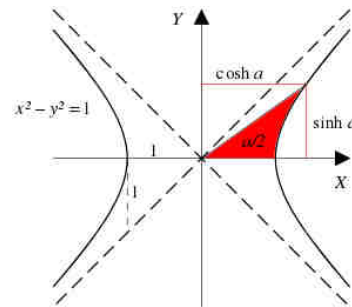
- 24-bit CORDIC rotation engine
 - Trigonometric and Hyperbolic functions
- Circular and Hyperbolic modes
- Rotation and Vectoring modes
- Programmable precision
- Low latency AHB slave interface
- DMA read and write channels



Cordic: Main features

261

- Supported functions:
 - Sine, Cosine: Vector rotation (polar to rectangular)
 - Arctanangent, Modulus: Vector translation (rectangular to polar)
 - Hyperbolic cosine
 - Hyperbolic sine
 - Hyperbolic tangent
 - Natural logarithm
 - Square root
- Fixed point signed input and output
 - Q1.31 or Q1.15 format
- Performance improvement relative to software
 - Sine and Cosine functions are typically 5x faster than software execution





CORDIC - Precision

262

- Precision is dependent on number of iterations

Function	Number of iteration (precision)	Number of cycles	Max residual error	
			q1.31 format	q1.15. format
Sin, Cos, Phase, Mod, Atan	4	1	2^{-3}	2^{-3}
	8	2	2^{-7}	2^{-7}
	12	3	2^{-11}	2^{-11}
	16	4	2^{-15}	2^{-15}
	20	5	2^{-18}	2^{-16}
	24	6	2^{-19}	2^{-16}



CORDIC - Precision

263

Function	Number of iteration (precision)	Number of cycles	Max residual error	
			q1.31 format	q1.15. format
Sinh, Cosh, Atanh, Ln	4	1	2^{-2}	2^{-2}
	8	2	2^{-6}	2^{-6}
	12	3	2^{-10}	2^{-10}
	16	4	2^{-13}	2^{-13}
	20	5	2^{-17}	2^{-15}
	24	6	2^{-18}	2^{-15}
Sqrt	4	1	2^{-7}	2^{-7}
	8	2	2^{-14}	2^{-14}
	12	3	2^{-19}	2^{-15}



CORDIC accelerator

264

Objective:

- To show how to configure the CORDIC accelerator.
 - Calculate the SIN and COS of angles with 6 cycles of precision.
- Enable the Event Recorder to measure the time for the CORDIC and ARM Math Library functions.
- Display CORDIC results in the Logic Analyzer.



CORDIC configuration

265

- Open CubeMX project “CORDIC_SinCos_DMA_Perf.ioc” file
 - \labs\Cordic\CORDIC_SinCos_DMA_Perf

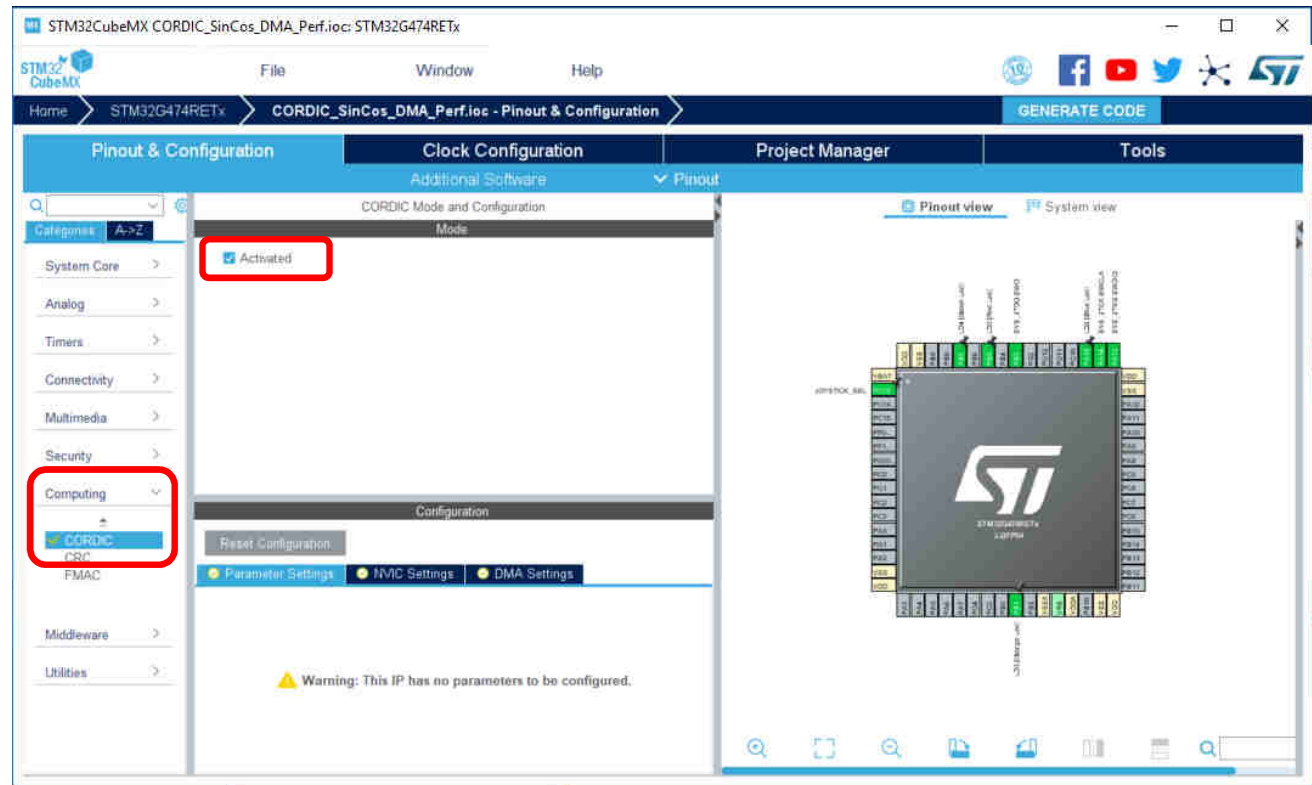
Name	Date modified	Type	Size
Drivers	8/28/2019 4:57 PM	File folder	
Inc	8/28/2019 8:22 PM	File folder	
MDK-ARM	8/29/2019 2:53 AM	File folder	
Src	8/28/2019 8:22 PM	File folder	
.mxproject	8/28/2019 8:22 PM	MXPROJECT File	7 KB
MX CORDIC_SinCos_DMA_Perf.ioc	8/29/2019 3:32 AM	IOC File	9 KB



CORDIC configuration

266

- Expand “Computing” Category
- Select “CORDIC”
 - CHECK “Activated”





CORDIC configuration

267

- DMA Request is enabled to transfer data in and out of CORDIC
- 32-bit Data Width (Word)

Configuration

Reset Configuration

Parameter Settings NVIC Settings **DMA Settings**

DMA Request	Channel	Direction	Priority
CORDIC_READ	DMA1 Channel 1	Peripheral To Memory	Low
CORDIC_WRITE	DMA1 Channel 2	Memory To Peripheral	Low

Add Delete

DMA Request Settings

Mode: Normal

Increment Address: ☐

Data Width: Word

DMA Request Synchronization Settings

Enable synchronization: ☐

Synchronization signal:

Signal polarity:

Enable event: ☐

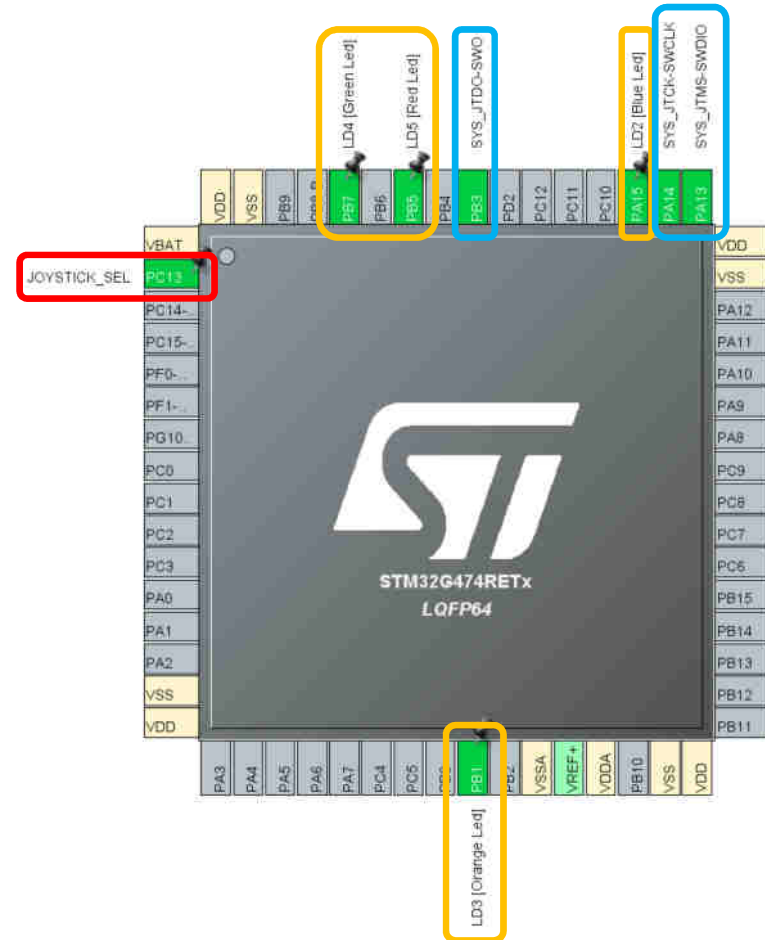
Request number:



Additional Peripheral configuration

268

- **JOYSTICK_SEL**: EXTI13
 - PC13
- **LD2-5**: GPIO_Output
 - PA15, PB1, PB5 and PB7
- **Serial Wire Debug with Trace (SWO)**:
 - PA13, PA14 and PB3

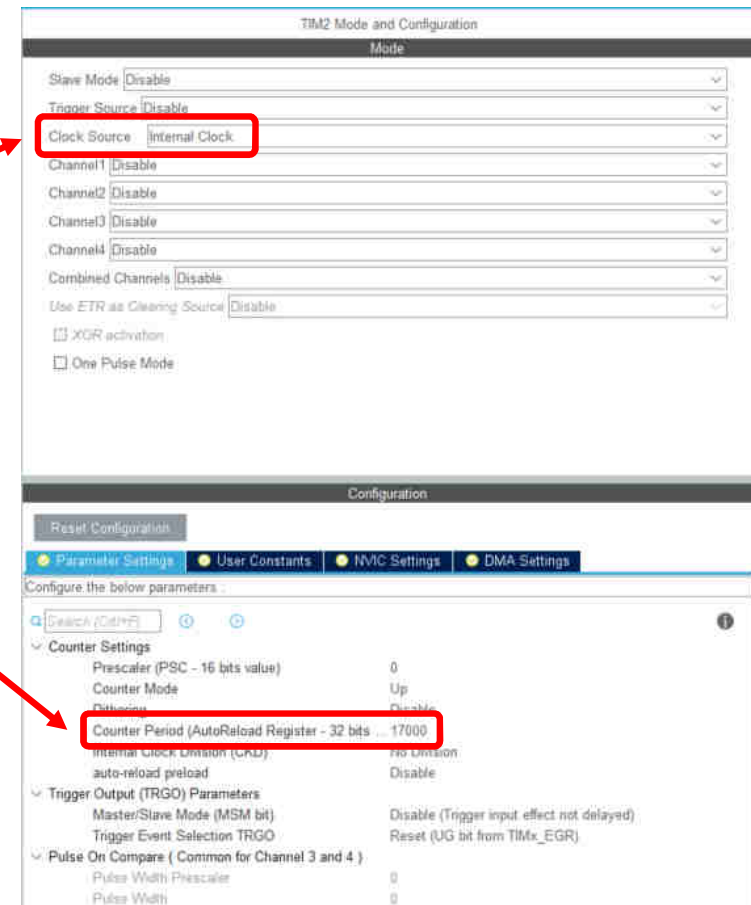
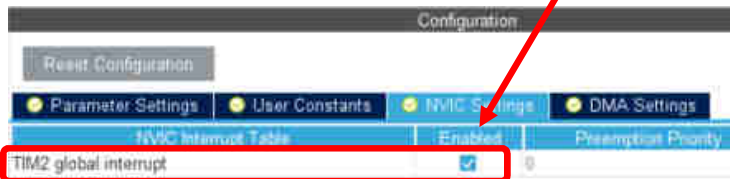




Additional Peripheral configuration

269

- **Timer 2:** Debug Logic Analyzer
- Internal Clock Source = 170 MHz
- Counter Period = 17000
 - 10 kHz timebase interrupt
- TIM2 global interrupt Enabled





270

- Validate “HCLK (MHz)” is set to 170

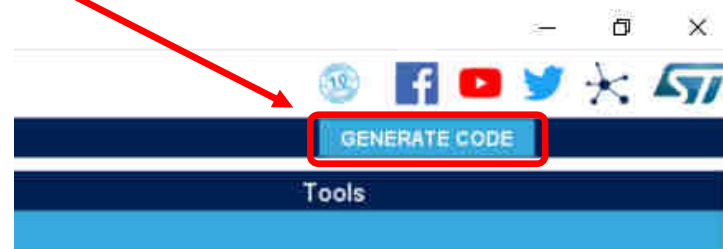




Hands-On: CORDIC

271

- Generate Code



- Open Project in MDK-ARM





Hands-On: CORDIC

272

- aAnglesCordic - Array of angles for CORDIC Q1.31 format (Line 57)

```
56 /* Array of angles for CORDIC Q1.31 format, regularly incremented from 0 to 2*pi */
57 static int32_t aAnglesCordic[ARRAY_SIZE] =
58 {
59     0x00000000, 0x04000000, 0x08000000, 0x0C000000,
```

- aAnglesLib – Array of angles for ARM Math library (Line 78)

```
77 /* Array of angles for arm math library Q1.31 format, regularly incremented from 0 to 2*pi */
78 static int32_t aAnglesLib[ARRAY_SIZE] =
79 {
80     0x00000000, 0x02000000, 0x04000000, 0x06000000,
```

- aRefSin & aRefCos – Array of Sin & Cos result reference (Line 99 & 120)

```
98 /* Array of reference sines in Q1.31 format */
99 static int32_t aRefSin[ARRAY_SIZE] =
100 {
101     0x00000000, 0x0C8BD35E, 0x18F8B83C, 0x25280C5D,
```

```
119 /* Array of reference cosines in Q1.31 format */
120 static int32_t aRefCos[ARRAY_SIZE] =
121 {
122     0x80000000, 0x7F62368F, 0x7D8A5F3F, 0x7A7D055B,
```

- aResults – Array of calculation results (Line 142)

```
140 /* Array of calculation results in Q1.31 format.
141    Will contain alternatively Sine and Cosine of input angles */
142 static int32_t aResults[2 * ARRAY_SIZE];
```




CORDIC configuration 273

- Sine calculation function (Line 219)
- 6 cycles of precision (Line 220)
- No Scaling (Line 221)
- One input angle data (Line 222)
 - Q1.31 format for input data (Line 224)
- Two output data (Sine and Cosine) (Line 223)
 - Q1.31 format for input data (Line 225)

```
217  /* USER CODE BEGIN 2 */
218  /*##-2- Configure the CORDIC peripheral #####
219  sCordicConfig.Function      = CORDIC_FUNCTION_SINE;
220  sCordicConfig.Precision     = CORDIC_PRECISION_6CYCLES;
221  sCordicConfig.Scale         = CORDIC_SCALE_0;
222  sCordicConfig.NbWrite       = CORDIC_NBWRITE_1;
223  sCordicConfig.NbRead        = CORDIC_NBREAD_2;
224  sCordicConfig.InSize        = CORDIC_INSIZE_32BITS;
225  sCordicConfig.OutSize       = CORDIC_OUTSIZE_32BITS;
```



CORDIC configuration

274

- Step = 0

- Calculation made with CORDIC
- Simple call to
HAL_CORDIC_Calculate_DMA

```
246  /*##### Calculation using CORDIC #####*/
247  case 0:
248      for (uint32_t i = 0; i < LOOP_NB; i++)
249      {
250
251          /* Start calculation of sines in DMA mode */
252          if (HAL_CORDIC_Calculate_DMA(&hcordic, aAnglesCordic, aResults,
253                                     ARRAY_SIZE, CORDIC_DMA_DIR_IN_OUT) != HAL_OK)
254          {
255              /* Processing Error */
256              Error_Handler();
257          }
258      }
```

- Step = 1

- Calculation made with ARM Math library
- Lines 277 through 283 are called to
perform same CORDIC operation

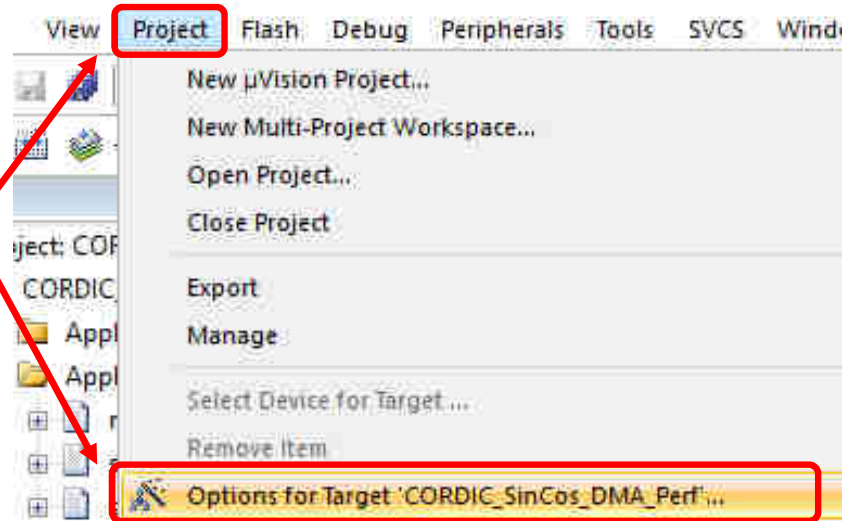
```
272  /*##### Calculation using arm math library #####*/
273  case 1:
274      for (uint32_t i = 0; i < LOOP_NB; i++)
275      {
276
277          for (uint32_t j = 0; j < ARRAY_SIZE; j++)
278          {
279              /* Calculate sine */
280              aResults[2*j] = arm_sin_q31(aAnglesLib[j]);
281
282              /* Calculate cosine */
283              aResults[(2*j) + 1] = arm_cos_q31(aAnglesLib[j]);
284          }
285      }
```



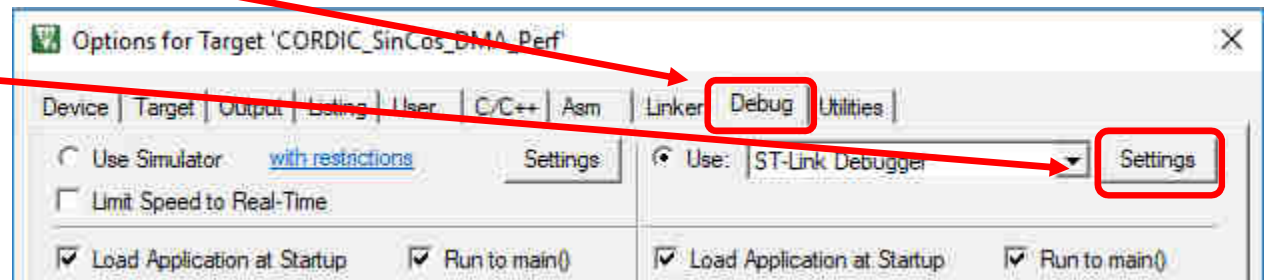
Enable Debug Trace

275

- Open “Options for Target ‘CORDIC_SinCos_DMA_Perf’ ...” window.



- Switch to “Debug” tab
- Click “Settings” button

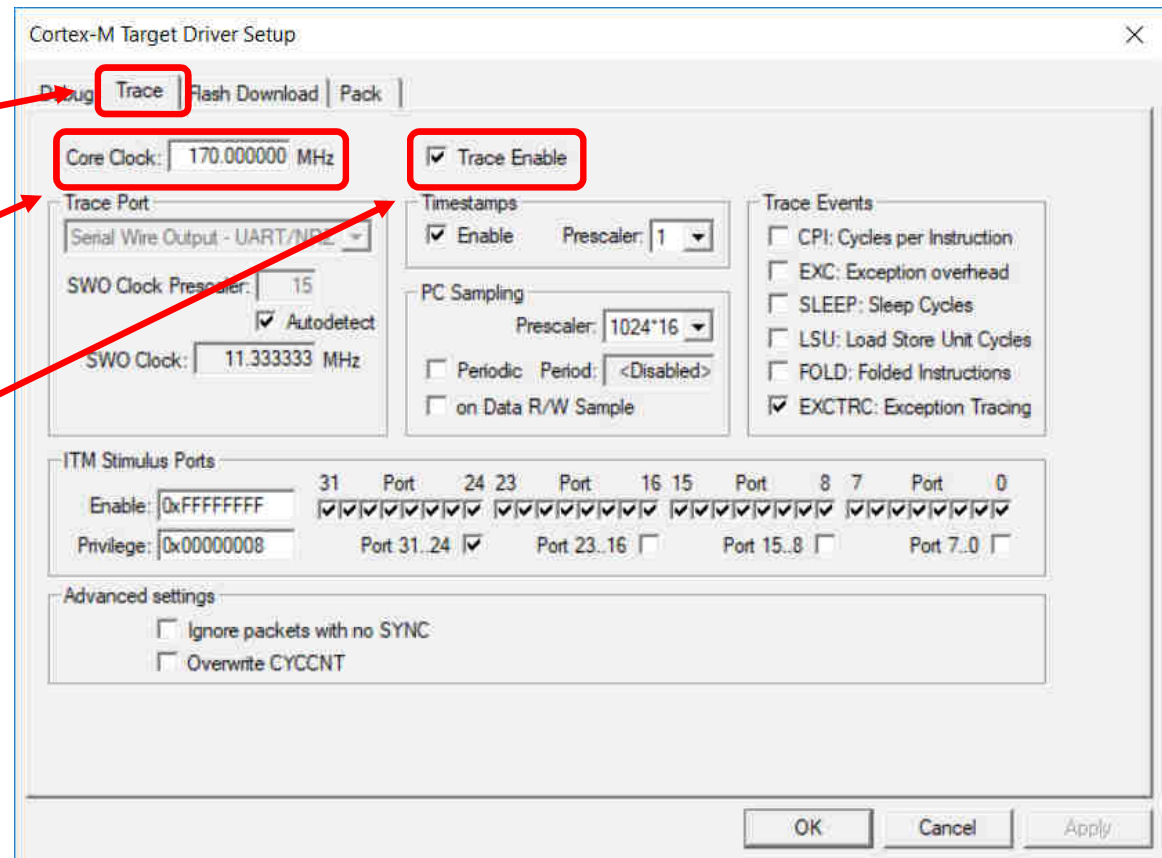




Enable Debug Trace

276

- Switch to “Trace” tab
- Set “Core Clock” = 170 MHz
- Click “Trace Enable” check box



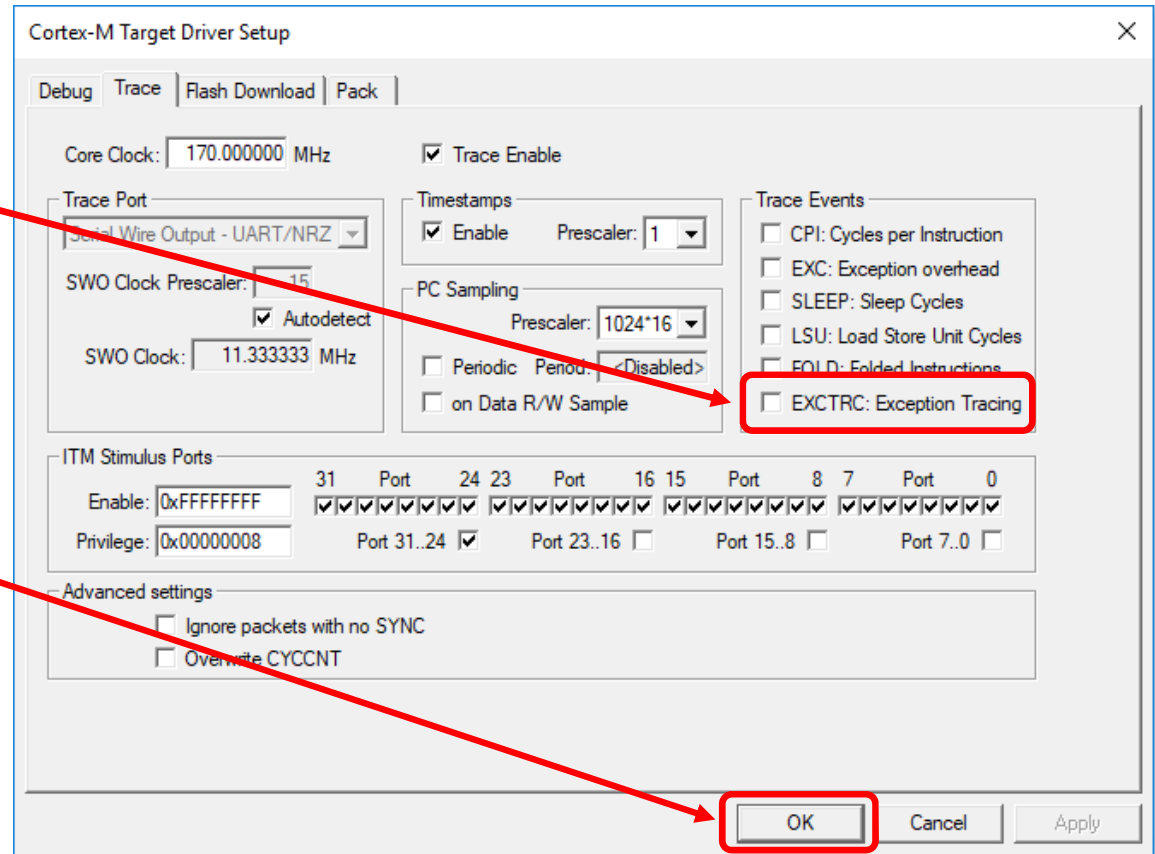


Enable Debug Trace

277

- Disable “Exception Tracing”
 - Uncheck Box

- Click “OK” to close setup window

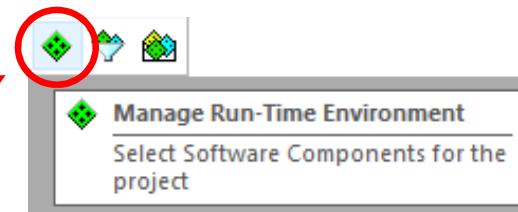




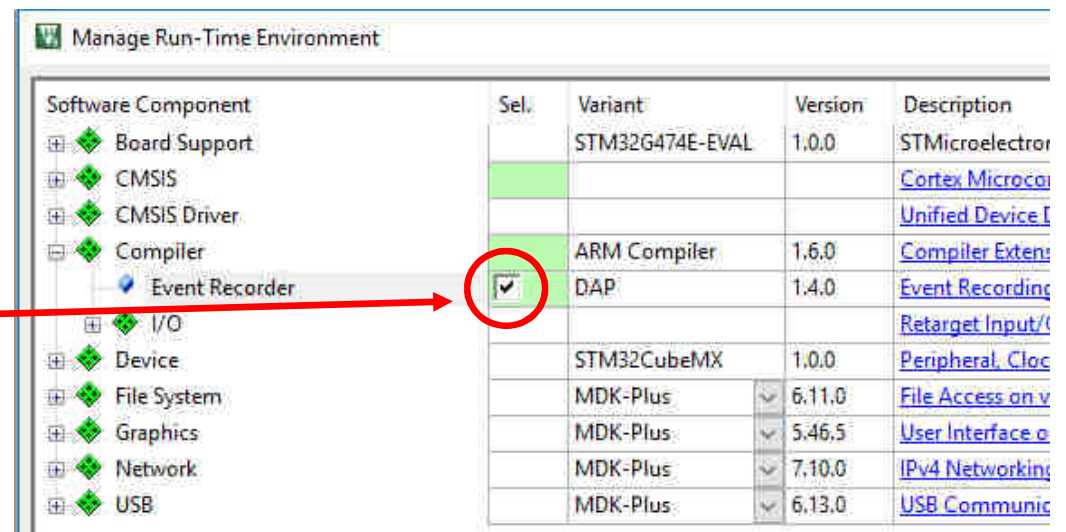
Enable Event Recorder

278

- Click on Manage Run-Time Environment



- Expand “Compiler”
- Enable “Event Recorder”
 - Check Box
- Click “OK”





Enable Event Recorder

279

- Open “main.c” under Application/User folder



- Line 27: Add (uncomment)

#include “EventRecorder.h”

```
24  /* Private includes -----
25  /* USER CODE BEGIN Includes */
26  #include <arm_math.h>
27  #include "EventRecorder.h"
28  /* USER CODE END Includes */
```

- Line 191 & 192: Add (uncomment)

EventRecorderInitialize(EventRecordAll, 1);
EventRecorderStart();

```
178 int main(void)
179 {
180     /* USER CODE BEGIN 1 */
181     /* STM32G4xx HAL library initialization:
182        - Configure the Flash prefetch
183        - SysTick timer is configured by default as source of time base, but user
184          can eventually implement his proper time base source (a general purpose
185          timer for example or other time source), keeping in mind that Time base
186          duration should be kept 1ms since PPP_TIMEOUT_VALUES are defined and
187          handled in milliseconds basis.
188        - Set NVIC Group Priority to 4
189        - Low Level Initialization
190     */
191     EventRecorderInitialize(EventRecordAll, 1);
192     EventRecorderStart();
193     /* USER CODE END 1 */
```



Enable Event Recorder

280

- Add (uncomment)

Line 250: EventRecord2(0,i,1);

Line 258: EventRecord2(0,i,2);

```
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
/*##### Calculation using CORDIC #####*/  
case 0:  
    for (uint32_t i = 0; i < LOOP_NB; i++)  
    {  
        EventRecord2(0,i,1);  
        /* Start calculation of sines in DMA mode */  
        if (HAL_CORDIC_Calculate_DMA(&hcordic, aAnglesCordic, aResults,  
                                     ARRAY_SIZE, CORDIC_DMA_DIR_IN_OUT) != HAL_OK)  
        {  
            /* Processing Error */  
            Error_Handler();  
        }  
        EventRecord2(0,i,2);  
    }
```

- Add (uncomment)

Line 276: EventRecord2(1,i,1);

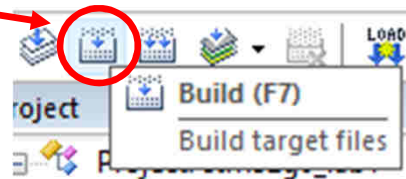
Line 285: EventRecord2(1,i,2);

```
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
/*##### Calculation using arm math library #####*/  
case 1:  
    for (uint32_t i = 0; i < LOOP_NB; i++)  
    {  
        EventRecord2(1,i,1);  
        for (uint32_t j = 0; j < ARRAY_SIZE; j++)  
        {  
            /* Calculate sine */  
            aResults[2*j] = arm_sin_q31(aAnglesLib[j]);  
  
            /* Calculate cosine */  
            aResults[(2*j) + 1] = arm_cos_q31(aAnglesLib[j]);  
        }  
        EventRecord2(1,i,2);  
    }
```

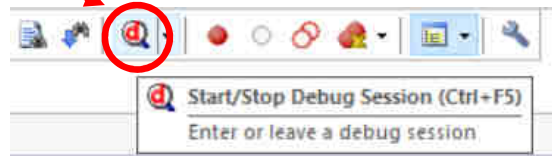

Hands-On: CORDIC

281

- Click the “Build”



- “Start Debug” Session



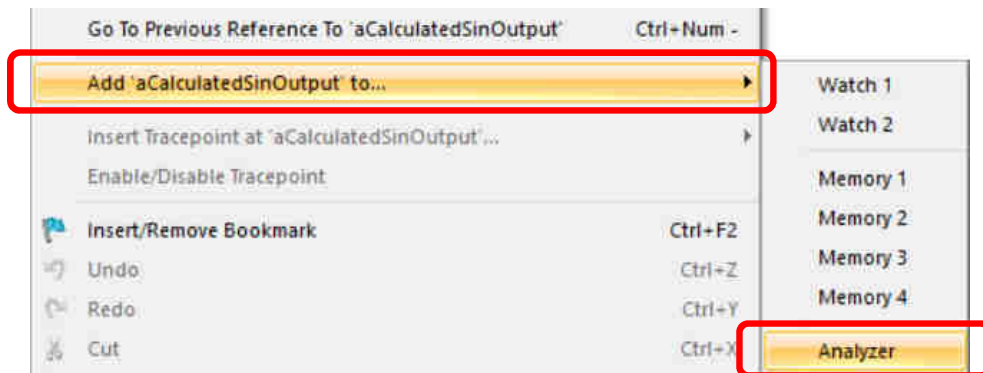
Enable Logic Analyzer

282

- On line 152, Right mouse click on “aCalculatedSinOutput” variable

```
151  /* Logic Analyzer variables */  
152  static int32_t aCalculatedSinOutput = 0;  
153  static int32_t aCalculatedCosOutput = 0;  
154  int32_t aResults_ptr = 0;
```

- Move mouse over “Add ‘aCalculatedSinOutput’ to...”
- Click on “Analyzer”



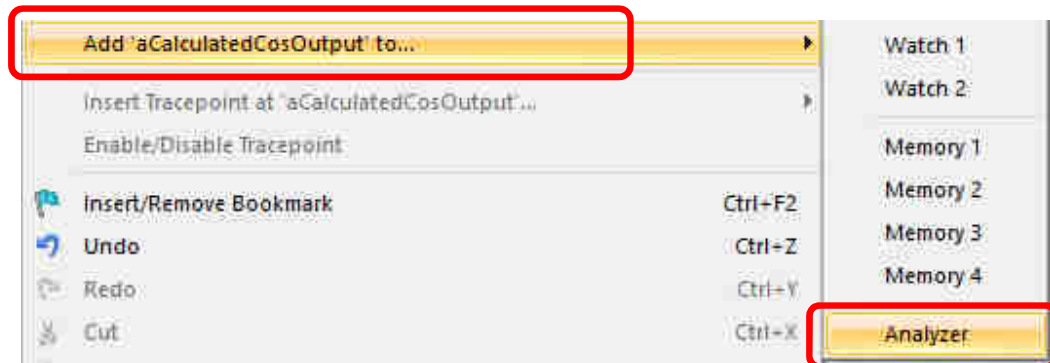
Enable Logic Analyzer

283

- On line 153, Right mouse click on “aCalculatedCosOutput” variable

```
151  /* Logic Analyzer variables */  
152  static int32 t aCalculatedSinOutput = 0;  
153  static int32_t aCalculatedCosOutput = 0;  
154  int32_t aResults_ptr = 0;
```

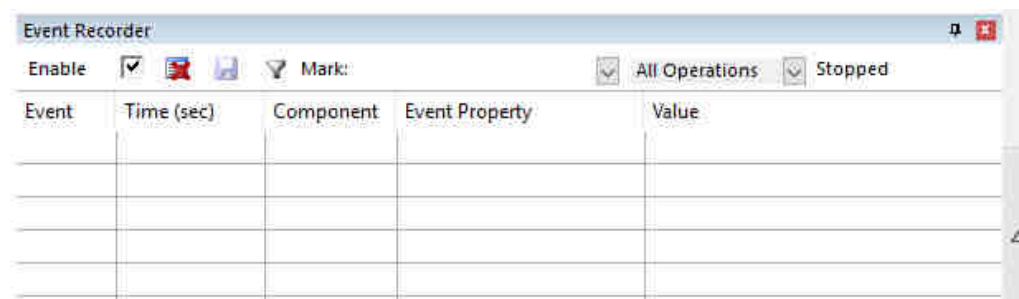
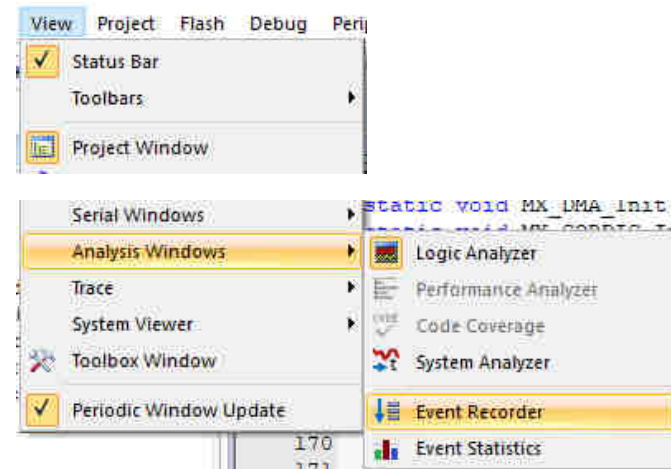
- Move mouse over “Add ‘aCalculatedCosOutput’ to...”
- Click on “Analyzer”



View Event Recorder

284

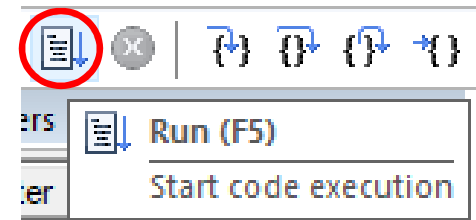
- Go to “View” menu
- Select “Analysis Windows”
- Select “Event Recorder”



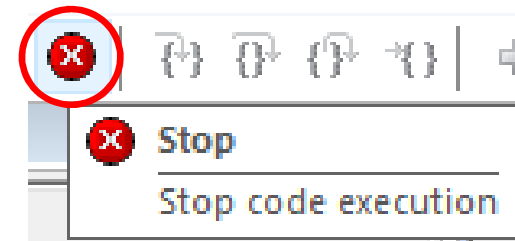
Run the Project

285

- “Run” the project
 - Capture trace data
 - Press “JOYSTICK_SEL” triggers “CORDIC” or “ARM Math library”
 - Green Led blinking
 - Fast blink (CORDIC used)
 - Slow blink (ARM Math Lib. Used)



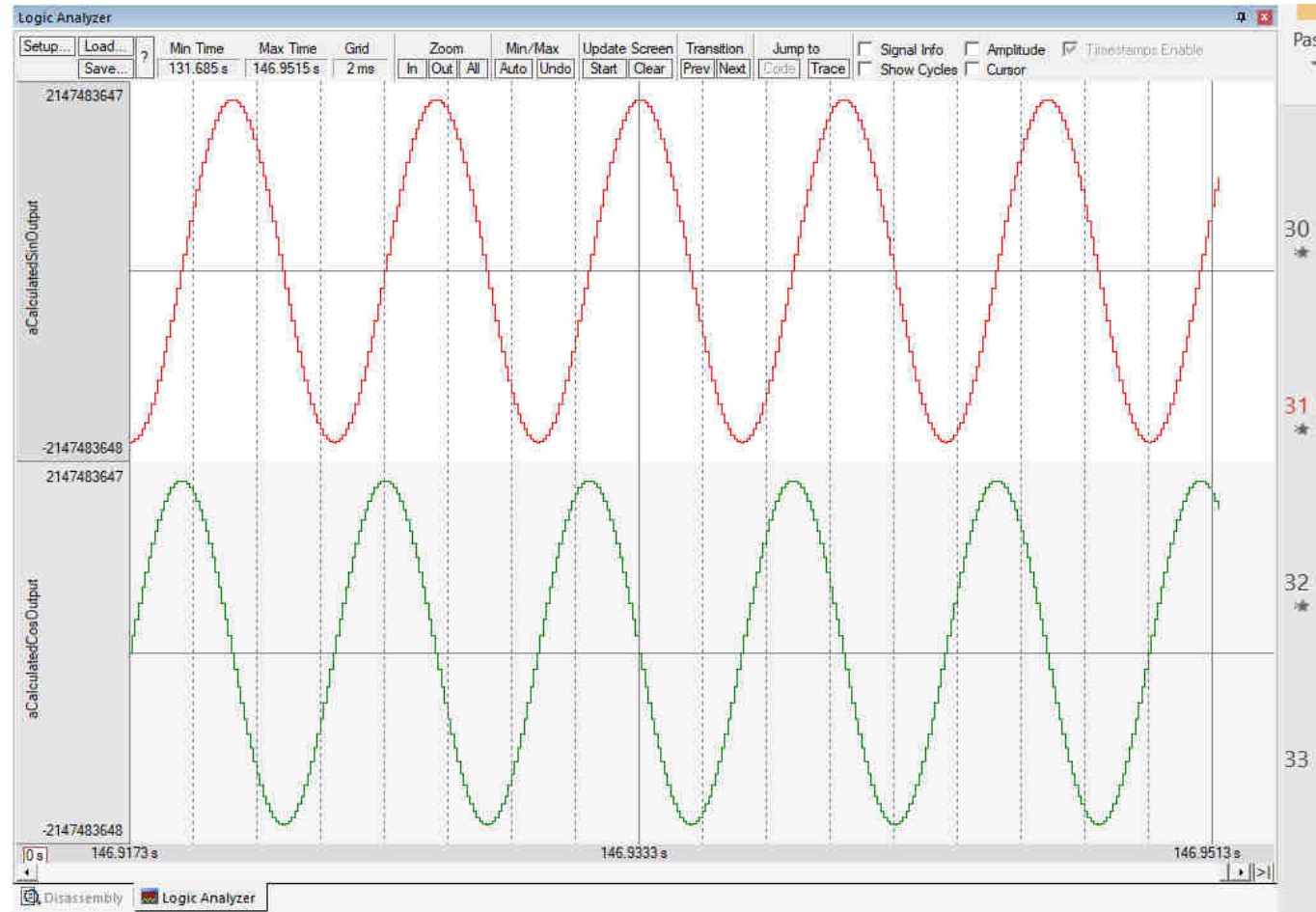
- “Stop” the project execution



Logic Analyzer

286

- CORDIC results are shown in Logic Analyzer



Event Recorder

287

- Events Captured

- Event Number
- Event Time Stamp
- Event Property
 - Id = 0 (CORDIC)
 - Id = 1 (ARM Math Library)

- Event Values

- 1st: Loop Number
- 2nd: Start(0x01) and End(0x02) of array calculation

Enable	☒	☐	☐	Mark:	☐	All Operations	☐	Stopped- Missed 4523
Event	Time (sec)	Component	Event Property	Value				
41819	1.44499562		id=0x0000	0x000031AF,0x00000002				
41820	1.44500605		id=0x0000	0x000031B0,0x00000001				
41821	1.44501337		id=0x0000	0x000031B0,0x00000002				
41822	1.44502308		id=0x0000	0x000031B1,0x00000001				
41823	1.44502995		id=0x0000	0x000031B1,0x00000002				
41824	1.44503989		id=0x0000	0x000031B2,0x00000001				
41825	1.44504672		id=0x0000	0x000031B2,0x00000002				
41826	1.44505666		id=0x0000	0x000031B3,0x00000001				
41827	1.44506349		id=0x0000	0x000031B3,0x00000002				
41828	1.44507343		id=0x0000	0x000031B4,0x00000001				
41829	1.44508026		id=0x0000	0x000031B4,0x00000002				
41830	1.44509020		id=0x0000	0x000031B5,0x00000001				
41831	1.44510062		id=0x0000	0x000031B5,0x00000002				
41832	1.55492708		id=0x0000	0x0000057F,0x00000001				
41833	1.55493392		id=0x0000	0x0000057F,0x00000002				

	Execution Time
CORDIC	6.84 us
ARM Math Library	42.24 us



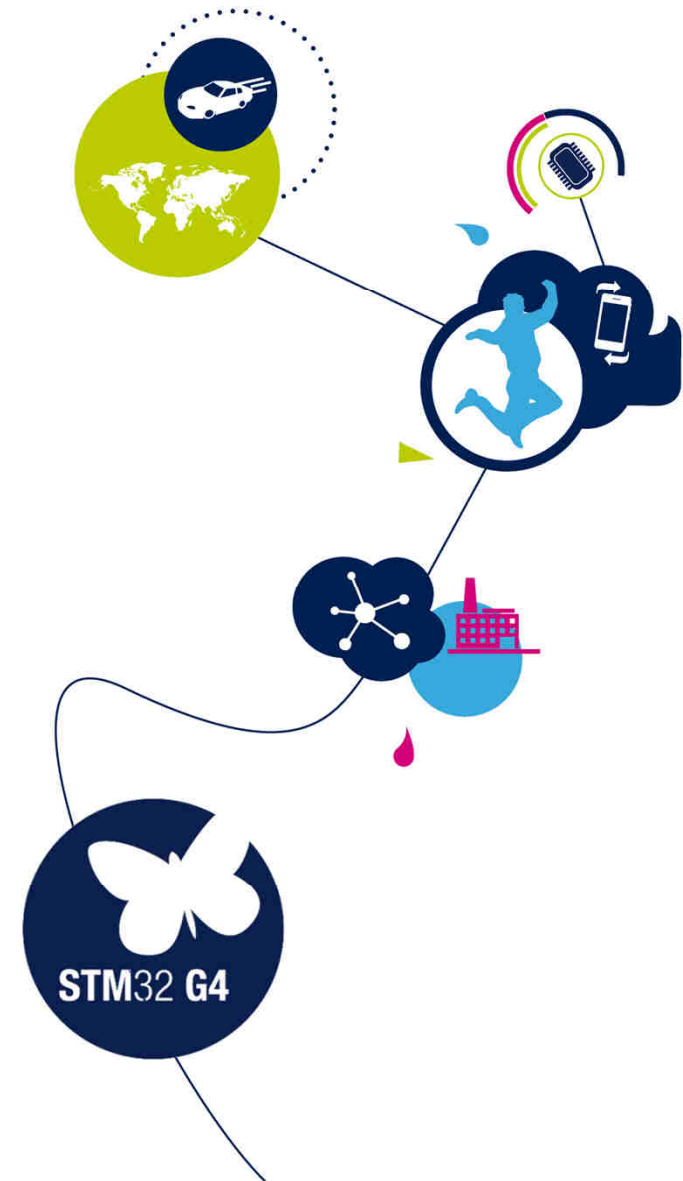
CORDIC: Additional Notes

288

- Use Cube Low Level driver to reduce HAL overhead
- AN5325: Getting started with the CORDIC accelerator using the STM32CubeG4 MCU Package
- Keil Application Note 321: Event Recorder Debugging with STM32G Cortex-M0/M0+

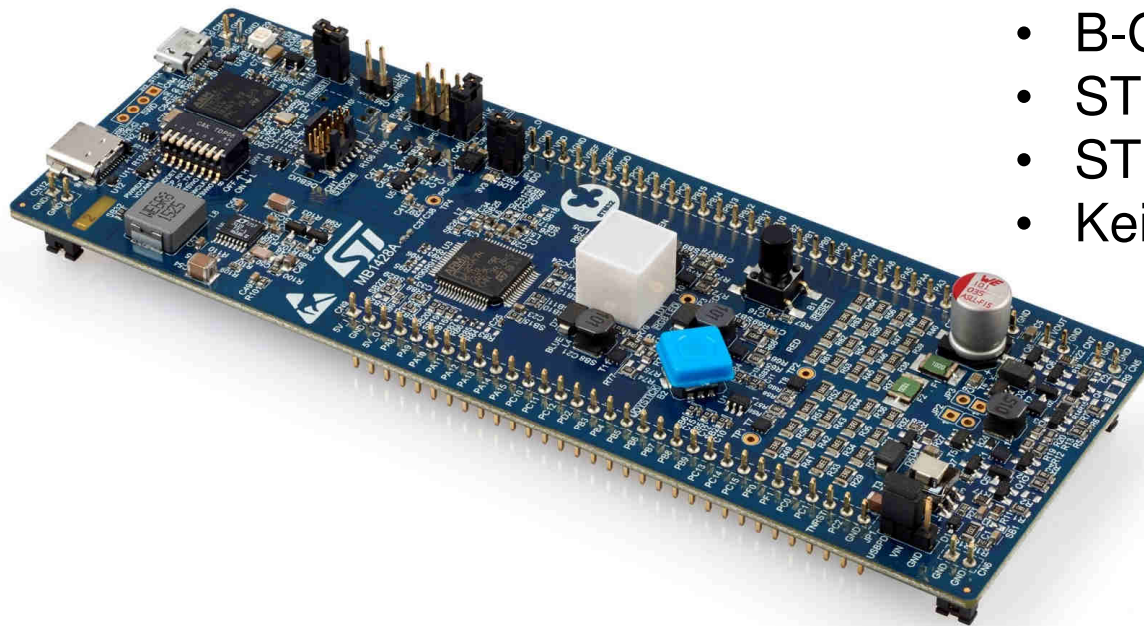
STM32G4 mixed-signal MCU workshop

*Hands On:
Filter Math Accelerator (FMAC)*





Systems Check



- USB cable
- B-G474E-DPOW1 Discovery Board
- STM32G4 Family Cube Library
- STM32CubeMx
- Keil MDK-ARM



Filter Math Accelerator

291

G4 FMAC

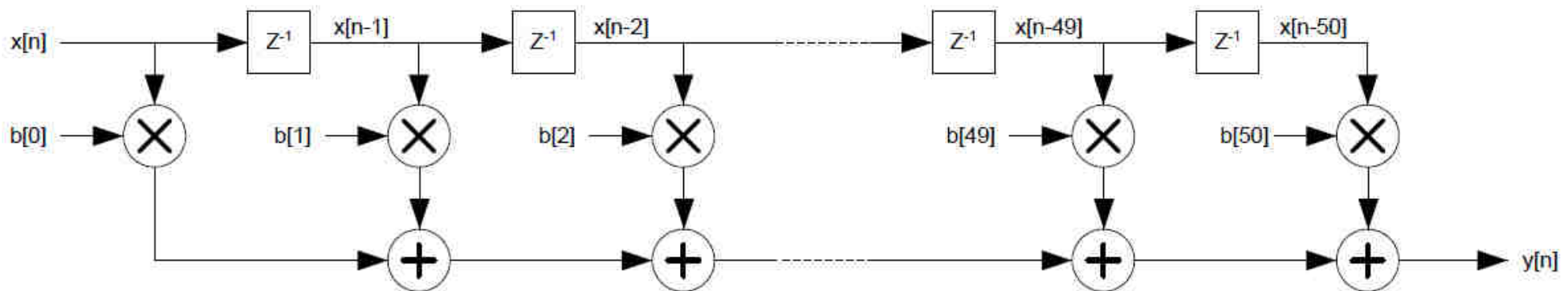
- Multiplier (16x16)
- Accumulator (24bit + 2bit)
- Address generation logic
- Input and Output Circular Buffers
- FIR & IIR filters possible
- Dot product, convolution, correlation
- Does not stall the CPU
- 256 x 16bit local memory
- Built-in sequencer
- Runs at the System Clock speed

Internally connected to:

- DMA
- AHB slave interface
- NVIC

G4 Adaptive FIR Filter

Figure 7. 51-tap FIR filter



- FIR filter is a convolution of the signal with filter coefficients
- The number of coefficients determines the minimum notch size
- Calculating N coefficients requires $N \times N$ operations



G4 Adaptive FIR Filter - Quantities

- Filter Coefficients will probably be in floating point format
- Coefficient conversion to 16 bit fixed point is necessary
- Fixed point format is Q1.15
 - Bits [0..14] represent fractional digits
 - Bit 15 is a sign bit
 - Range = -1.0 (0x8000) to +0.999969482421875 (0x7FFF)
- Proper design will keep the absolute value coefficients ≤ 1
 - If not, all coefficients will need to be divided by the greatest coefficient
- To convert floating to fixed point, the Floating Point Unit is used
 - FPU takes 8 cycles per conversion
 - 51 coefficients take ~408 clock cycles



Filter Math Accelerator

294

Lab steps 1 of 6

This Hands On Lab illustrates noise cancellation using an adaptive FIR filter.

The experiment will be run on two input frames:

- a gaussian noise with two tone interferers.
- a gaussian noise with three tone interferers.

Create individual coefficient sets.

The result is a "whitened" noise.

Figure 1. Gaussian noise with two-tone interferer

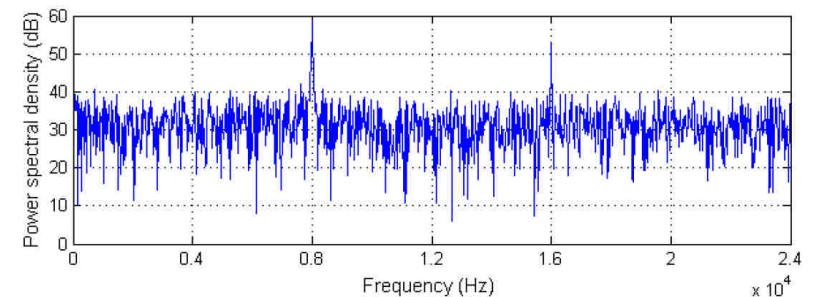
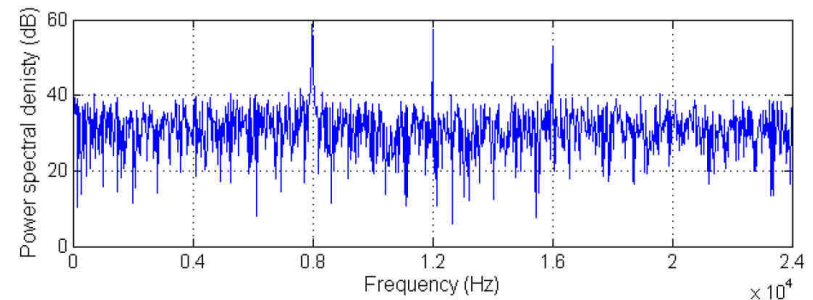


Figure 4. Gaussian noise with three-tone interferer





Filter Math Accelerator

295

Lab steps 2 of 6

- The FMAC peripheral will be configured to perform an adaptive FIR filter.
- There will be 51 feed-forward taps.
- The gain is equal to 0.
- The input and output thresholds are set to 1.
- The clipping feature can be enabled or disabled thanks to the compilation

switch `CLIP_ENABLED` in `main.h`.



Filter Math Accelerator

296

Lab steps 3 of 6

The FMAC peripheral is making the best use of its 256 x 16-bit internal memory.

The local memory is divided into three buffers:

- 0 to 50: coefficient buffer;
- 51 to 150: input buffer;
- 151 to 250: output buffer;
- 251 to 255: unused.



Filter Math Accelerator 297

Lab steps 4 of 6

- DMA mode is used to transfer input data from memory TO the FMAC peripheral.
- DMA mode is used to transfer output data FROM FMAC peripheral to memory.
- The coefficient buffer contains the vector of feed-forward taps
- This is named the B buffer
 - (size comprised in [2:127]; here, size $N = 51$ elements).
- This buffer is initialized during the configuration step in DMA mode.



Filter Math Accelerator

298

Lab steps 5 of 6

The size of the input buffer is the sum of:

- the size of the coefficient vector B ($N = 51$ elements);
- the size of the additional space D1 needed for throughput improvement (minimum: 0; here, $D1 = 49$ elements).

The input buffer is filled during the preload step in DMA mode; new values are added in DMA mode once the preloaded ones have been consumed.

The size of the output buffer is the one of the additional space D2 needed for a better throughput (minimum: 1; here, $D2 = 100$ elements).

The output buffer is read in DMA mode.



Filter Math Accelerator

299

Lab steps 6 of 6

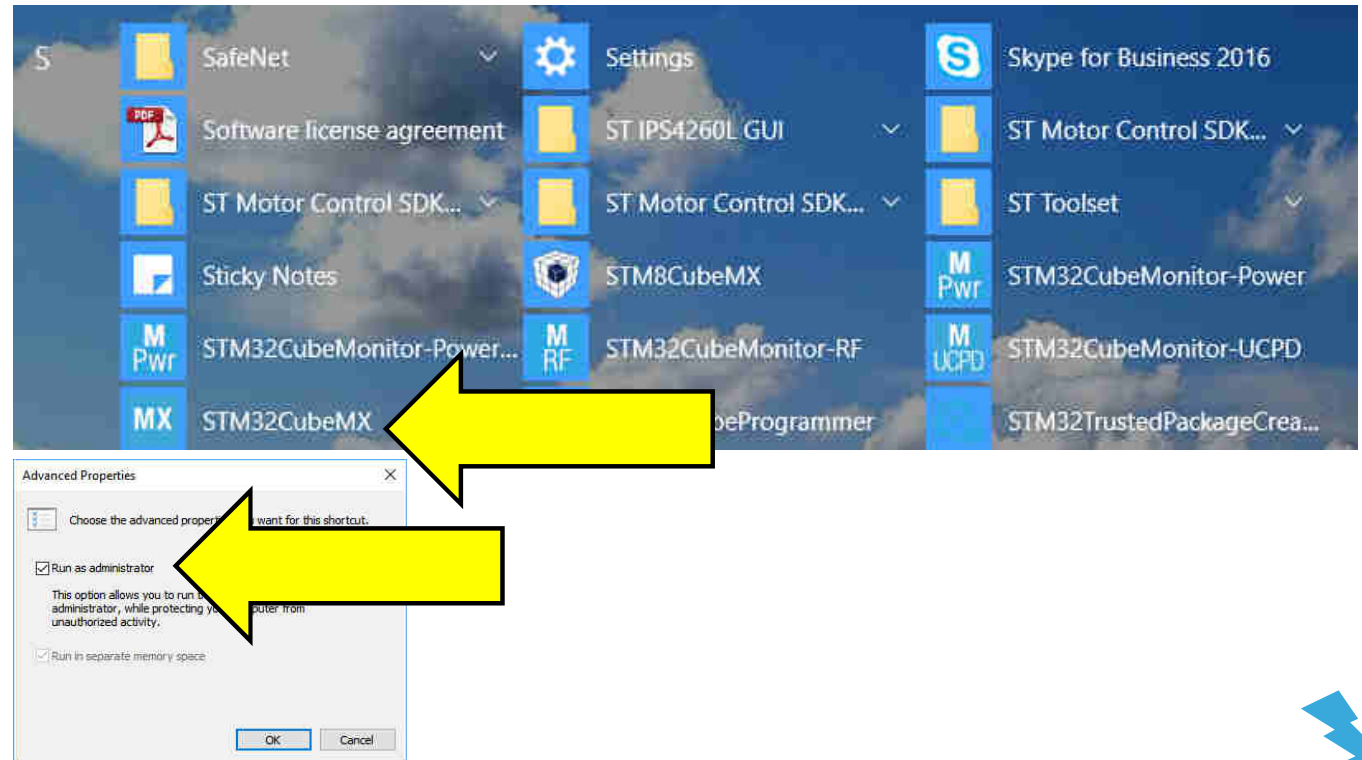
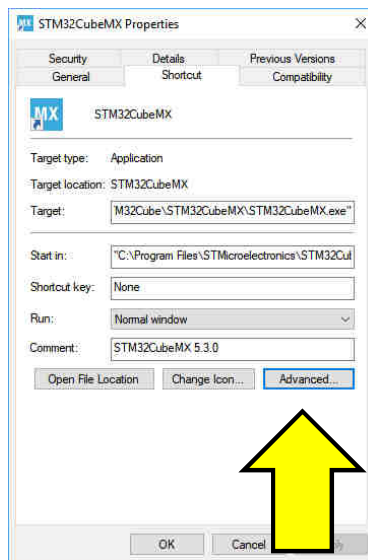
STM32 board LEDs are used to monitor the example status:

- LED4 is ON when correct FMAC FIR results have been calculated.
- LED5 is blinking (1 second period) if an error is detected in the FIR filter results or if there is an initialization or configuration error.

The filtered output is displayed in the Terminal I/O if the compilation switch `PRINT_OUTPUT` is enabled in `main.h`.



Start up





Splash Page – G4 Library Installed?

The screenshot displays the STM32CubeMX software interface. On the left, the 'Embedded Software Packages Manager' window is open, showing a table of installed and available software packages. A yellow arrow points to the 'STM32G4' section of the table. The main splash page on the right features a 'Manage software installations' section with a 'CHECK FOR UPDATES' button and an 'INSTALL / REMOVE' button. A large yellow arrow points from the 'INSTALL / REMOVE' button to the right, with the text 'Install/Remove' written inside it. Below the splash page, there is a section for the 'New multicore STM32MP1 Series for Industrial and IoT applications' featuring an STM32MP1 microcontroller and the OpenSTLinux Distribution logo.

Description	Installation	Available Version
STM32F7		
STM32G0		
STM32G4		
STM32Cube MCU Package for STM32G4 Series	1.1.0	1.1.0
STM32Cube MCU Package for STM32G4 Series	1.0.1	1.0.1

Details

From Local ... From Url ... Refresh Install Now Remove Now Close

Manage software installations

Check for STM32CubeMX and embedded software packa...

CHECK FOR UPDATES

Install or remove embedded software package

INSTALL / REMOVE

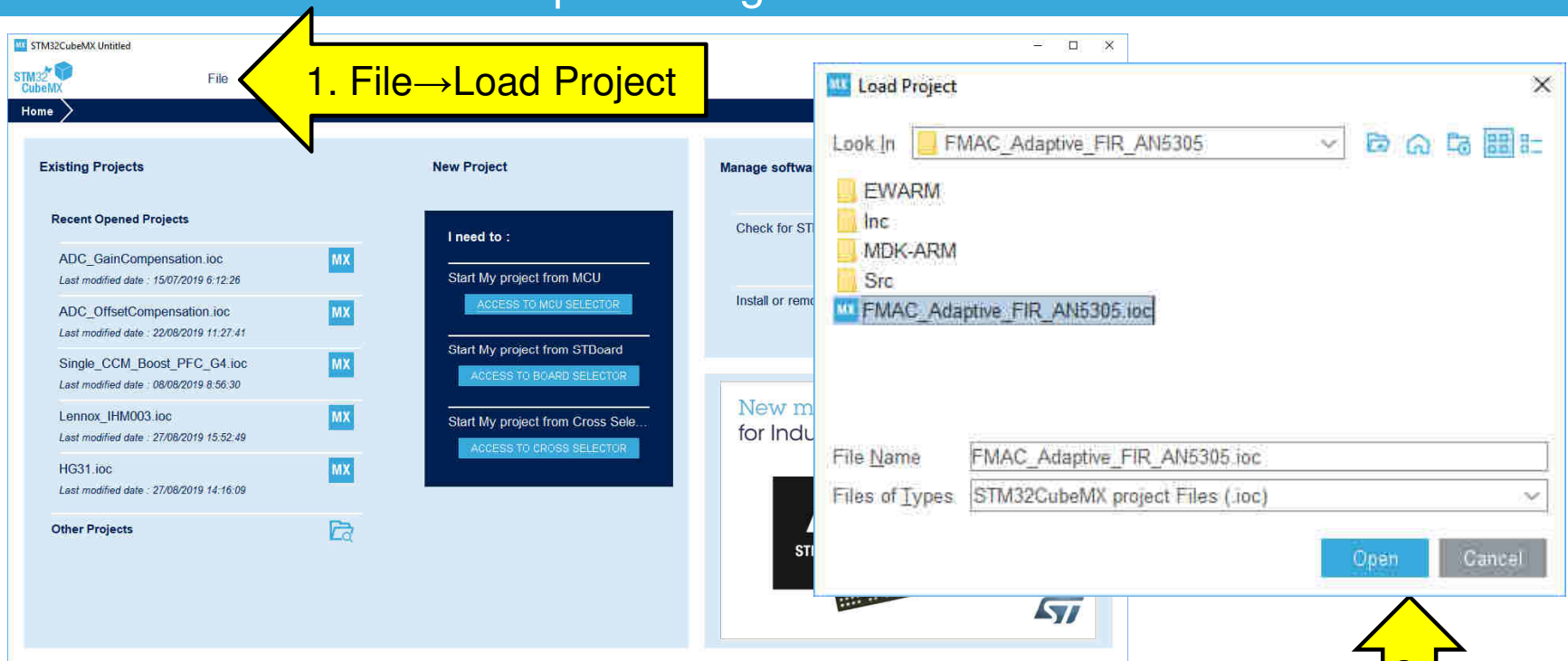
New multicore STM32MP1 Series for Industrial and IoT applications

STM32MP1

OpenSTLinux Distribution



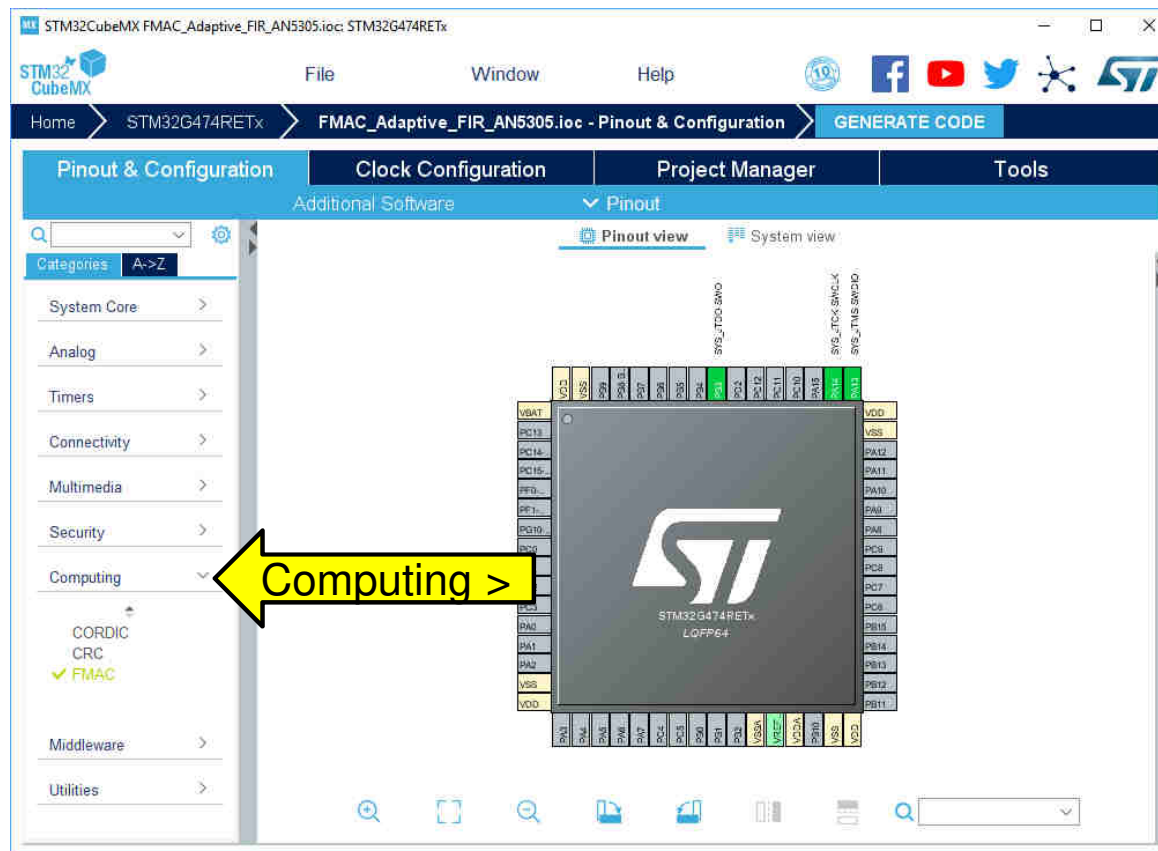
Splash Page



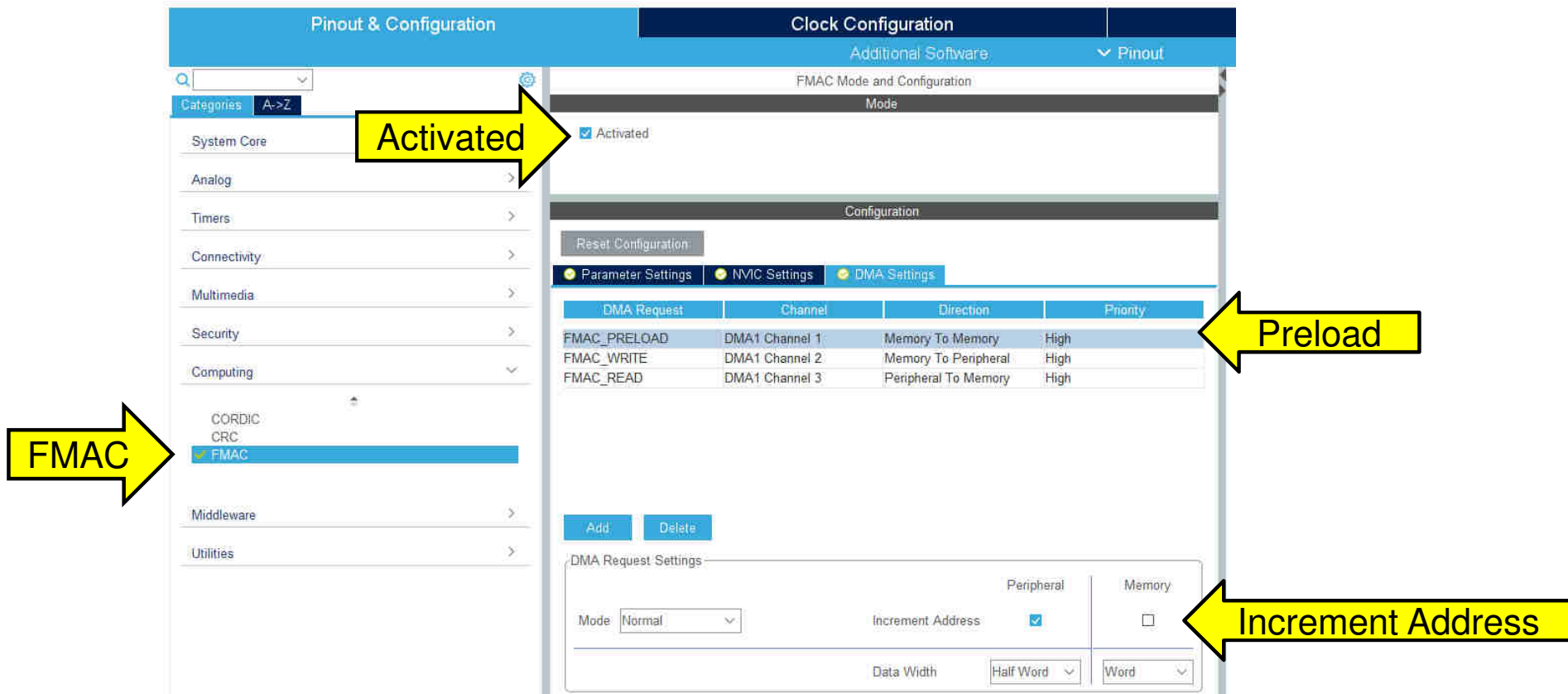
C:\...\STM32Cube\Repository\STM32Cube_FW_G4_V1.1.0\Projects\
B-G474E-DPOW1\Examples\FMAC\FMAC_Adaptive_FIR_AN5305



Graphic Design Representation



FMAC – 3 DMA Channels



Activated

FMAC

Preload

Increment Address

DMA Request	Channel	Direction	Priority
FMAC_PRELOAD	DMA1 Channel 1	Memory To Memory	High
FMAC_WRITE	DMA1 Channel 2	Memory To Peripheral	High
FMAC_READ	DMA1 Channel 3	Peripheral To Memory	High

DMA Request Settings:

Mode:

Increment Address: ☒

Data Width:



FMAC – 3 DMA Channels

Activated

FMAC

WRITE

Increment Address

DMA Request	Channel	Direction	Priority
FMAC_PRELOAD	DMA1 Channel 1	Memory To Memory	High
FMAC_WRITE	FMAC_WRITE	Memory To Peripheral	High
FMAC_READ	DMA1 Channel 3	Peripheral To Memory	High

DMA Request Settings

Mode: Normal

Increment Address: ☒

Data Width: Word



FMAC – 3 DMA Channels

Pinout & Configuration

Search

Categories: A-Z

- System Core
- Analog
- Timers
- Connectivity
- Multimedia
- Security
- Computing
 - CORDIC
 - CRC
 - FMAC**
- Middleware
- Utilities

Activated

Mode ☒ Activated

Configuration

Reset Configuration

Parameter Settings NVIC Settings **DMA Settings**

DMA Request	Channel	Direction	Priority
FMAC_PRELOAD	DMA1 Channel 1	Memory To Memory	High
FMAC_WRITE	DMA1 Channel 2	Memory To Peripheral	High
FMAC_READ	DMA1 Channel 3	Peripheral To Memory	High

READ

Add Delete

DMA Request Settings

Mode: Normal

Increment Address: ☐

Peripheral: ☐ Memory: ☒

Data Width: Word Half Word

Increment Address



Project Manager – Keil MDK-ARM V5

STM32CubeMX FMAC_Adaptive_FIR_AN5305.ioc: STM32G474RETx

File Window Help

Home > STM32G474RETx > FMAC_Adaptive_FIR_AN5305.ioc - Project Manager > GENERATE CODE

Pinout & Configuration Clock Configuration **Project Manager** Tools

Project

Project Settings

Project Name
FMAC_Adaptive_FIR_AN5305

Project Location
effery blausen\STM32Cube\Repository\STM32Cube_FW_G4_V1.1.0\Projects\B-G474E-DPOW1\Examples\FMAC

Application Structure
Basic ☐ Do not generate the main()

Toolchain Folder Location
pository\STM32Cube_FW_G4_V1.1.0\Projects\B-G474E-DPOW1\Examples\FMAC\FMAC_Adaptive_FIR_AN5305\

Toolchain / IDE
MDK-ARM V5

Code Generator

Advanced Settings

Linker Settings

Minimum Heap Size 0x200

Minimum Stack Size 0x400

Mcu and Firmware Package

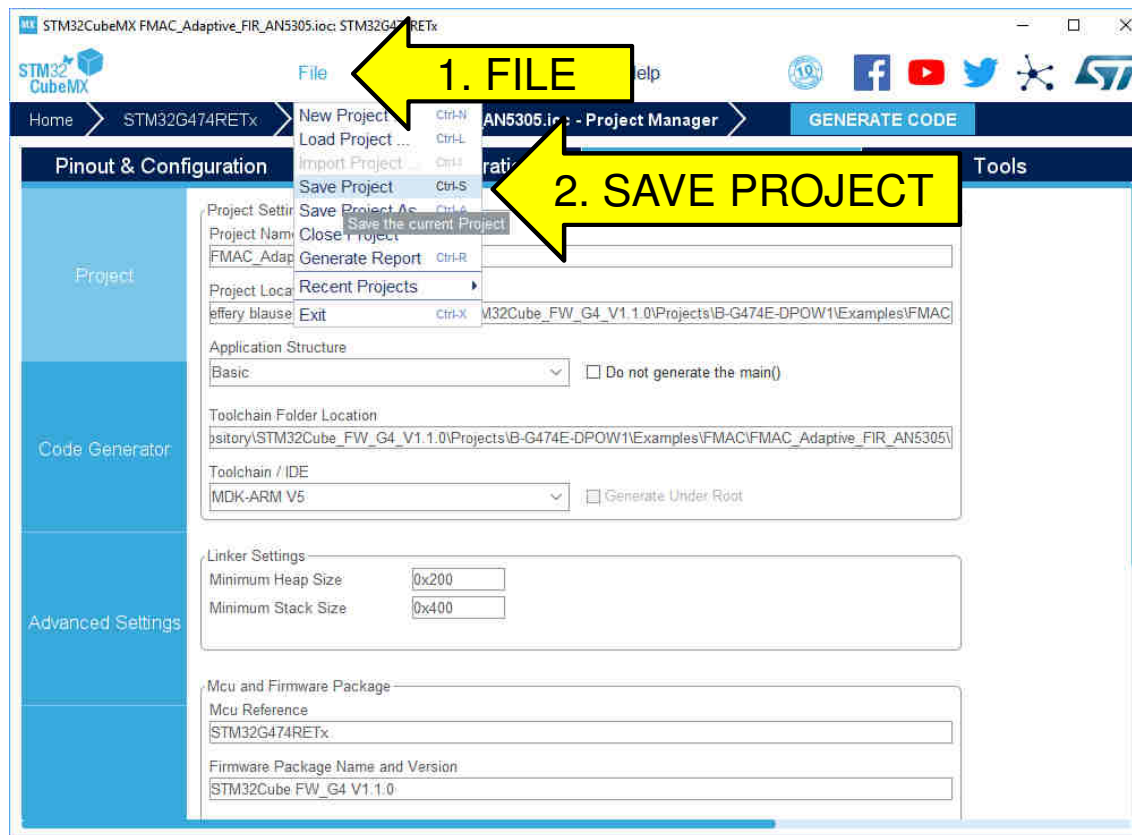
Mcu Reference
STM32G474RETx

Firmware Package Name and Version
STM32Cube FW_G4 V1.1.0

MDK-ARM V5

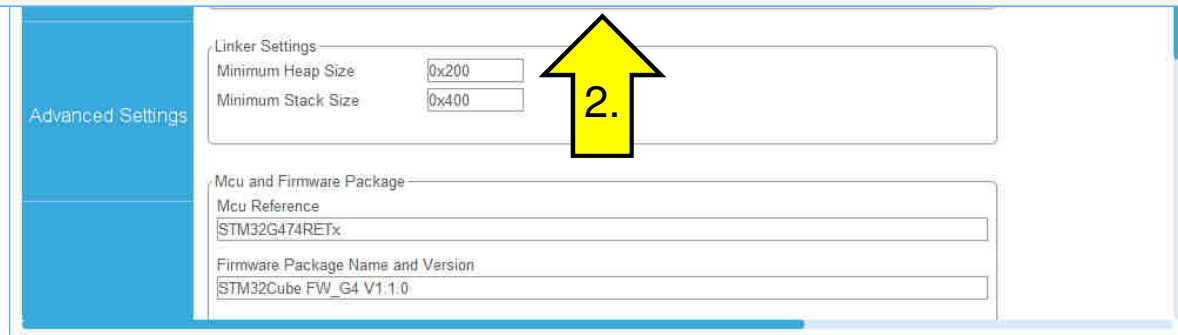
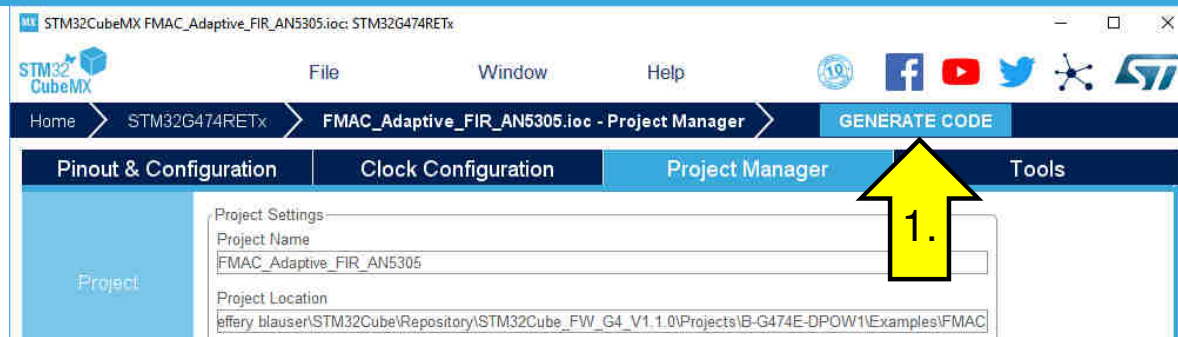


FILE→SAVE PROJECT





Generate Code





Keil Start up page

The screenshot shows the Keil MDK-ARM IDE interface. The left pane displays the project tree for 'Project: FMAC_Adaptive_FIR_AN5305'. The main editor window shows the 'main.c' file, which contains the startup code for the STM32G474xx. The code is as follows:

```
102
103 /**
104  * @brief The application entry point.
105  * @retval int
106  */
107 int main(void)
108 {
109     /* USER CODE BEGIN 1 */
110     /* STM32G4xx HAL library initialization:
111      - Configure the Flash prefetch
112      - SysTick timer is configured by default as source of time base, but user
113      can eventually implement his proper time base source (a general purpose
114      timer for example or other time source), keeping in mind that Time base
115      duration should be kept 1ms since PFP_TIMEOUT_VALUES are defined and
116      handled in milliseconds basis.
117      - Set NVIC Group Priority to 4
118      - Low Level Initialization
119     */
120     /* USER CODE END 1 */
121
122     /* MCU Configuration-----*/
123
124     /* Reset of all peripherals, Initializes the Flash interface and the SysTick. */
125     HAL_Init();
126
127     /* USER CODE BEGIN Init */
128
129     /* USER CODE END Init */
130
131     /* Configure the system clock */
132     SystemClock_Config();
133
134     /* USER CODE BEGIN SysInit */
135     /* Configure LED4 and LED5 */
136     BSP_LED_Init(LED4);
137     BSP_LED_Init(LED5);
138
139     /* Configure unused LED2 and LED3 */
140     BSP_LED_Init(LED2);
141     BSP_LED_Init(LED3);
142     BSP_LED_Off(LED2);
143     BSP_LED_Off(LED3);
144     /* USER CODE END SysInit */
145
146 }
```

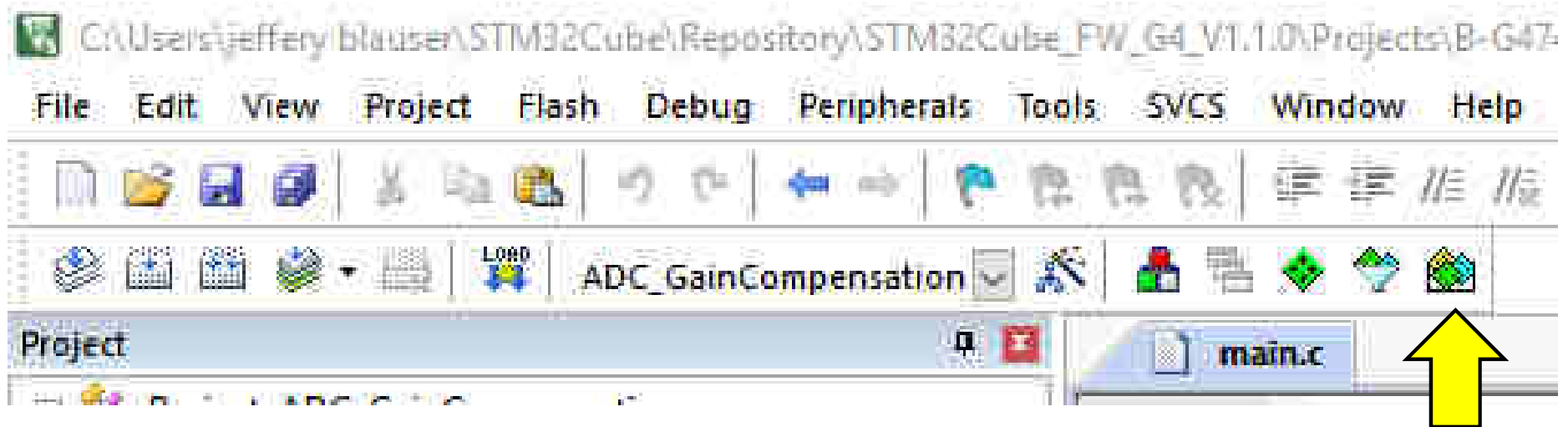




Keil MDK-ARM

311

Check the Device Pack





G4 Device Pack

Exit

STM32G474

Keil:STM32G4xx_DFP

Output

Cannot download file http://mcuxpresso.nxp.com/cmsis_pack/repo/NXP.LPC54018M_DFP.pdsc: Illegal MIME type 'text/html'

Cannot download file http://mcuxpresso.nxp.com/cmsis_pack/repo/NXP.LPC54018_DFP.pdsc: Illegal MIME type 'text/html'

Cannot download file http://mcuxpresso.nxp.com/cmsis_pack/repo/NXP.LPC54101_DFP.pdsc: Illegal MIME type 'text/html'

Action [2 left]: Update Pack descriptions, download http://mcuxpresso.nxp.com/cmsis_pack/repo/NXP.LPC54102_DFP.pdsc

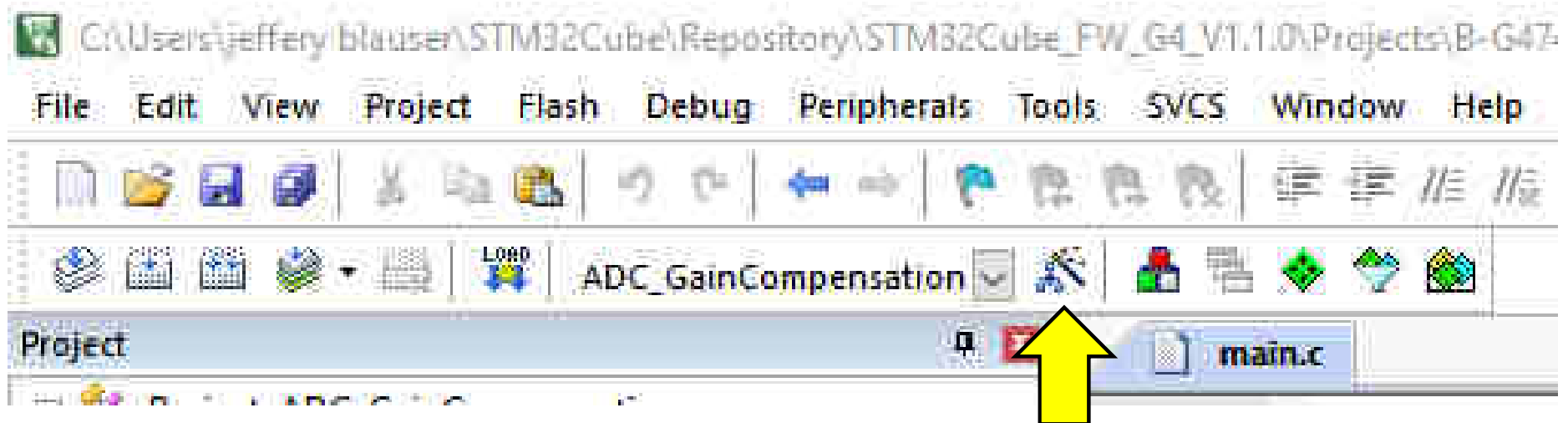
0% ONLINE



Keil MDK-ARM

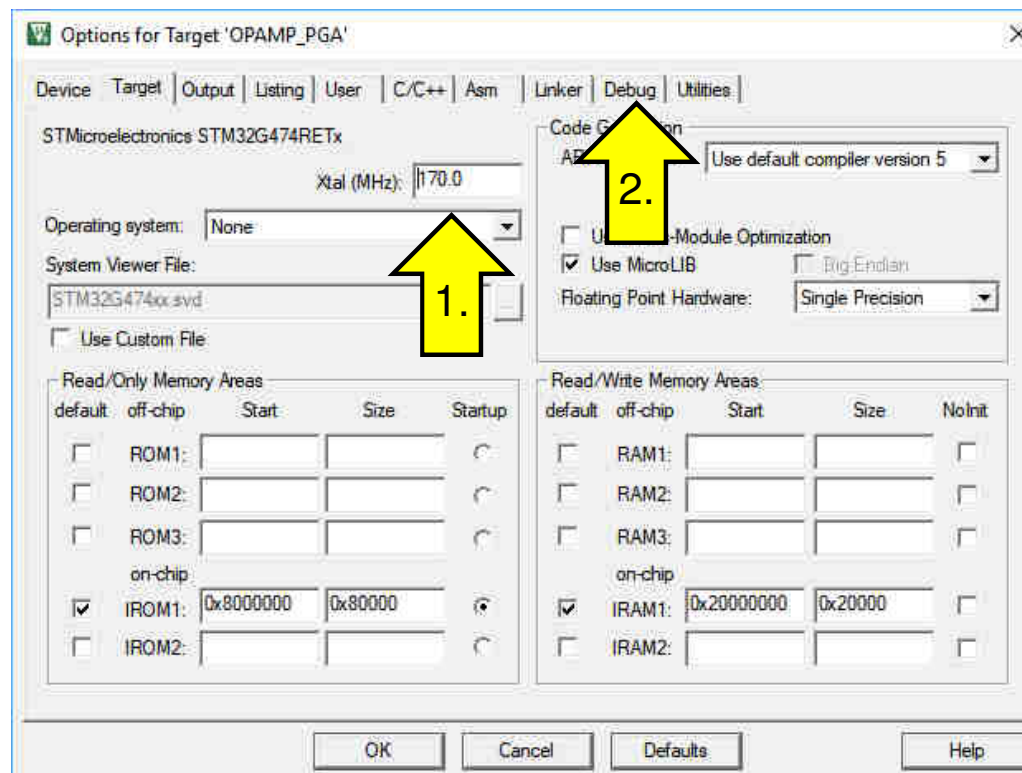
313

Check the Options



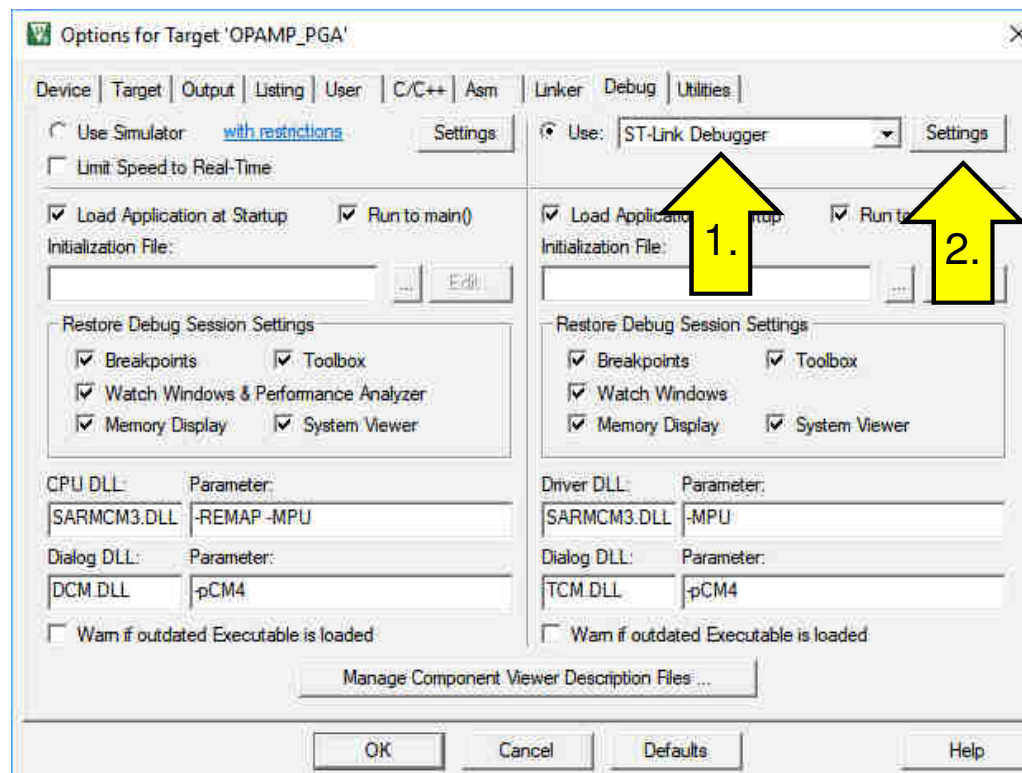


Clock speed and Debug



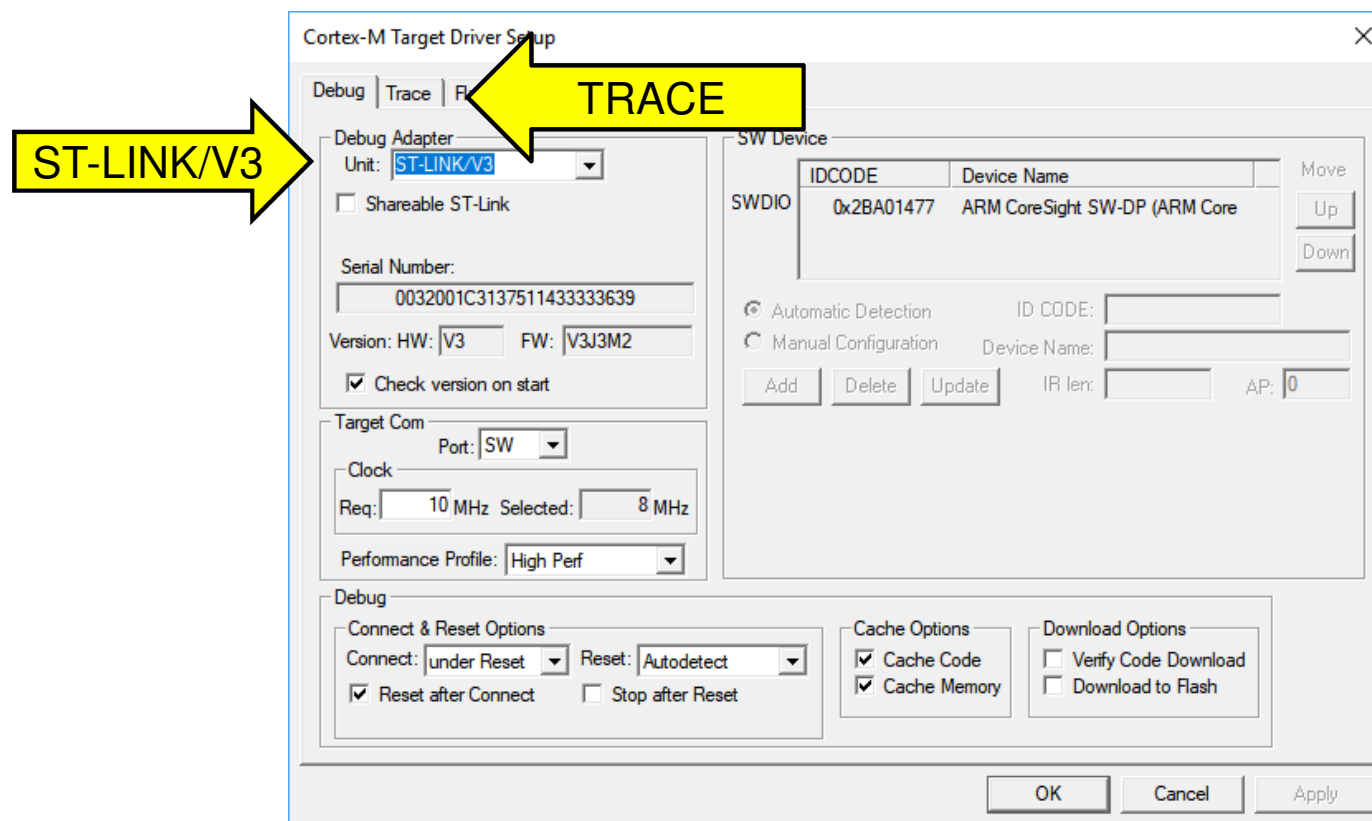


ST-Link Debugger - Settings



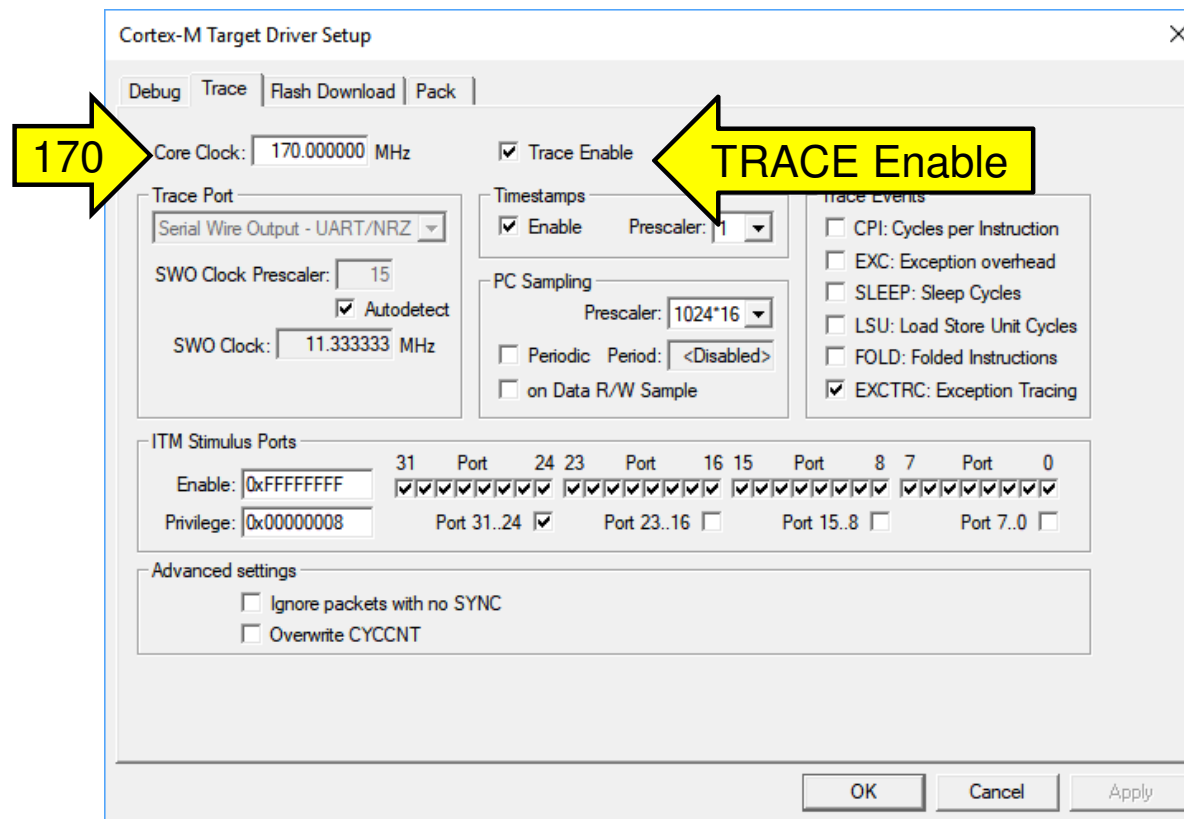


ST-Link/V3 & TRACE





Core Clock & TRACE Enable (Before)





Core Clock & TRACE Enable (After)

The screenshot shows the 'Cortex-M Target Driver Setup' dialog box with the 'Trace' tab selected. The following annotations highlight specific settings:

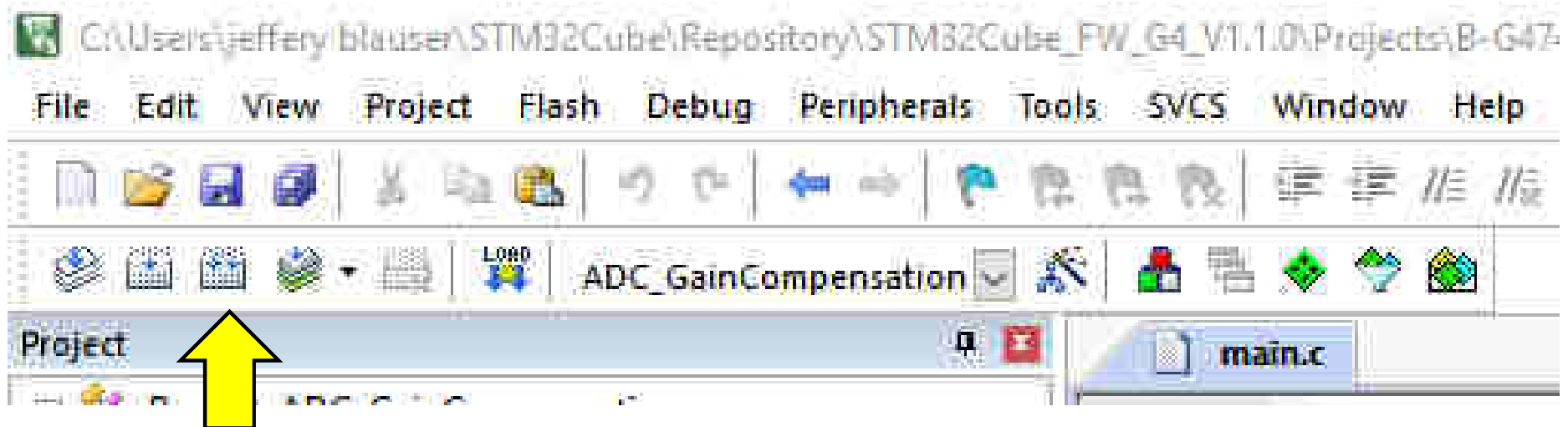
- 1. 170**: Points to the 'Core Clock' field, which is set to 170.000000 MHz.
- 2. TRACE Enable**: Points to the 'Trace Enable' checkbox, which is checked.
- 3. Uncheck EXCTRC**: Points to the 'EXCTRC: Exception Tracing' checkbox in the 'Trace Events' section, which is unchecked.
- 4. Uncheck all but 15**: Points to the 'ITM Stimulus Ports' section, where only port 15 is checked in the 'Enable' row.
- 5. OK**: Points to the 'OK' button at the bottom right of the dialog.

Additional visible settings include:

- Trace Port: Serial Wire Output - UART/NRZ
- SWO Clock Prescaler: 15
- SWO Clock: 11.333333 MHz
- Autodetect: checked
- Timestamps: checked
- PC Sampling: Prescaler: 1024*16
- ITM Stimulus Ports: Enable: 0x00008000, Privilege: 0x00000008
- Advanced settings: Ignore packets with no SYNC, Overwrite CYCCNT



Build

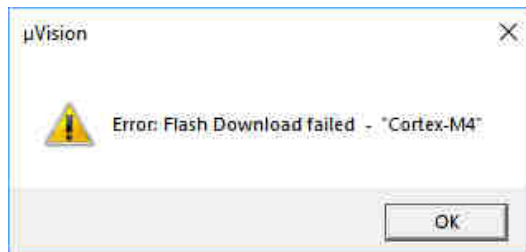
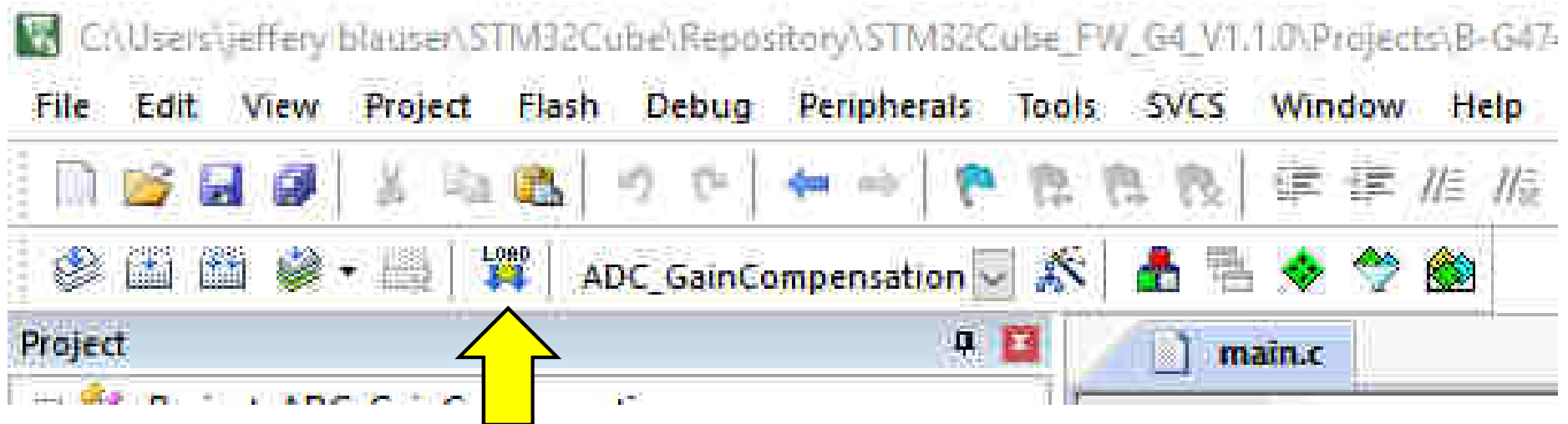


Build Output

```
Program Size: Code=7876 RO-data=8972 RW-data=116 ZI-data=13804
FromELF: creating hex file...
FMAC_Adaptive_FIR_AN5305\FMAC_Adaptive_FIR_AN5305.hex: Warning: Q9931W: Your license for feature Keil will expire in 29 days
Finished: 0 information, 1 warning and 0 error messages.
"FMAC_Adaptive_FIR_AN5305\FMAC_Adaptive_FIR_AN5305.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:01:00
```



Load



Build Output

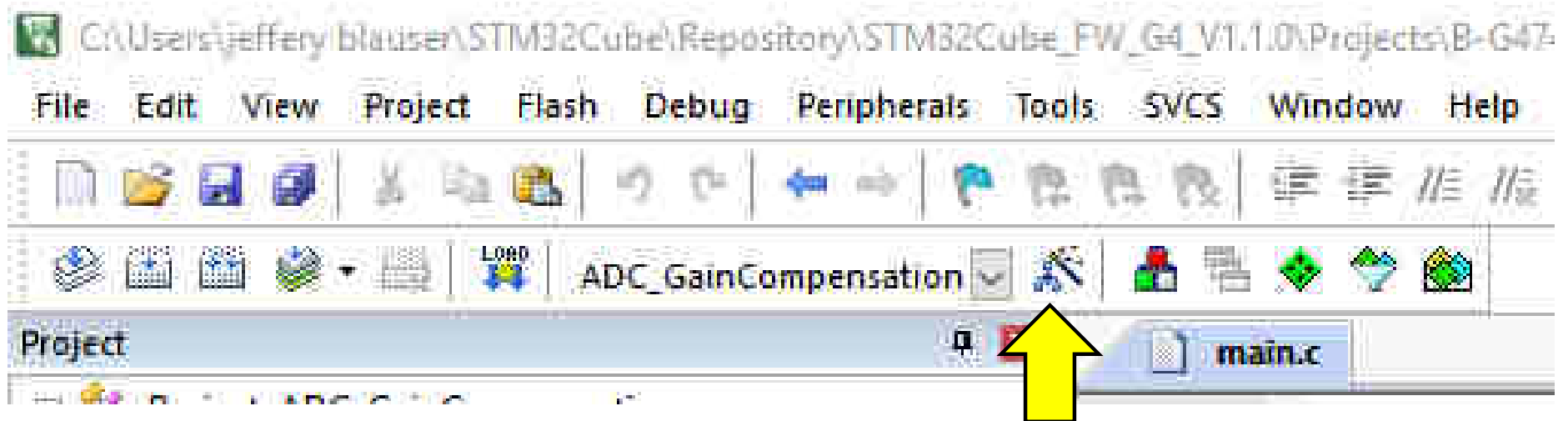
```
Contents mismatch at: 08001860H (Flash=38H Required=C5H) !
Contents mismatch at: 08001861H (Flash=46H Required=F8H) !
Contents mismatch at: 08001862H (Flash=FEH Required=88H) !
Contents mismatch at: 08001863H (Flash=F7H Required=00H) !
Contents mismatch at: 08001864H (Flash=B9H Required=60H) !
Contents mismatch at: 08001865H (Flash=FCH Required=6CH) !
Too many errors to display !
Error: Flash Download failed - "Cortex-M4"
```




Keil MDK-ARM

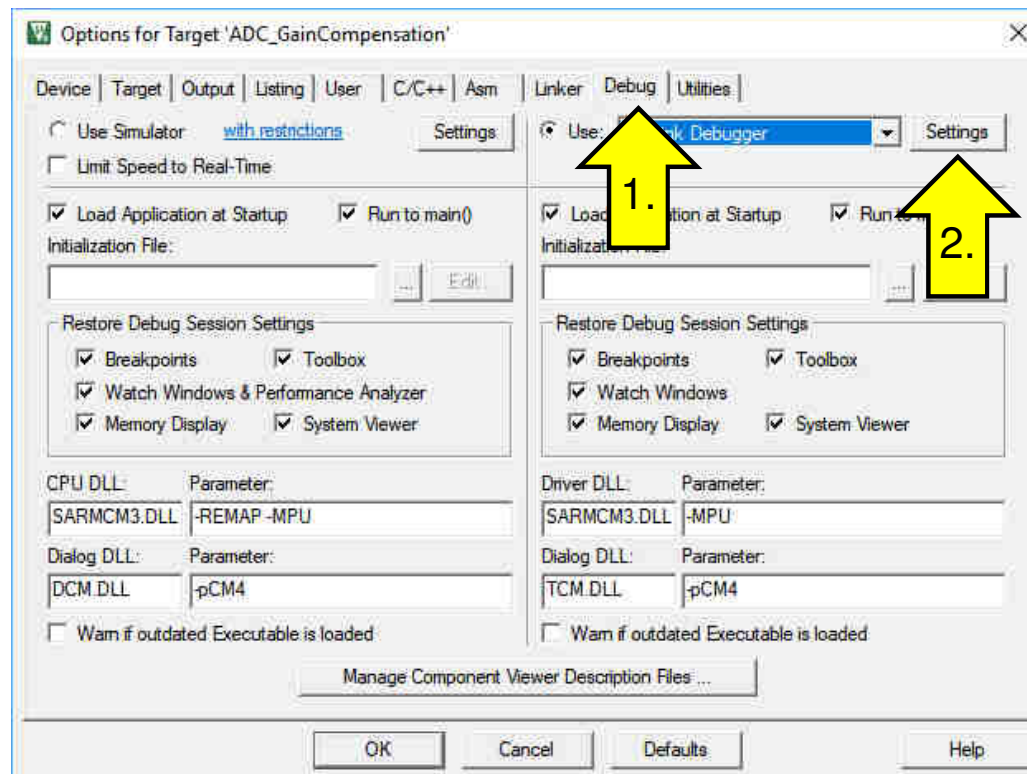
321

Options



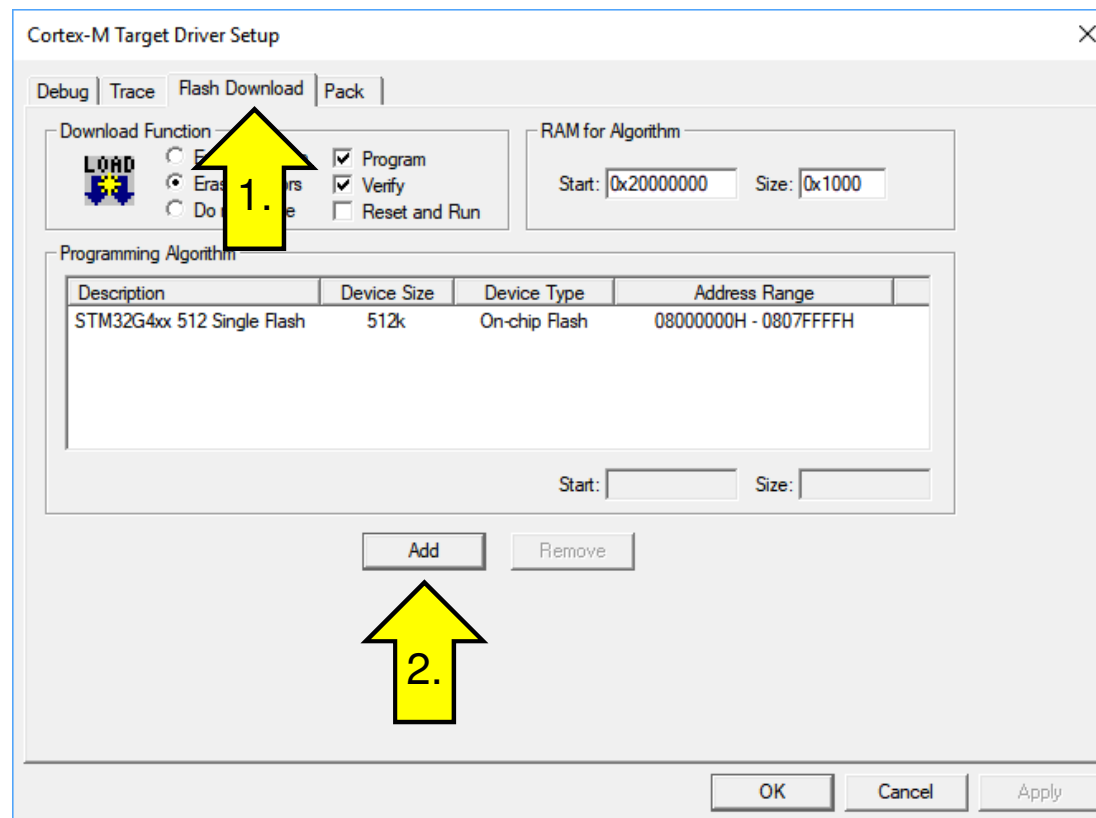


ST-Link Debugger - Settings



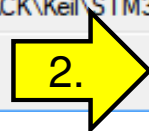
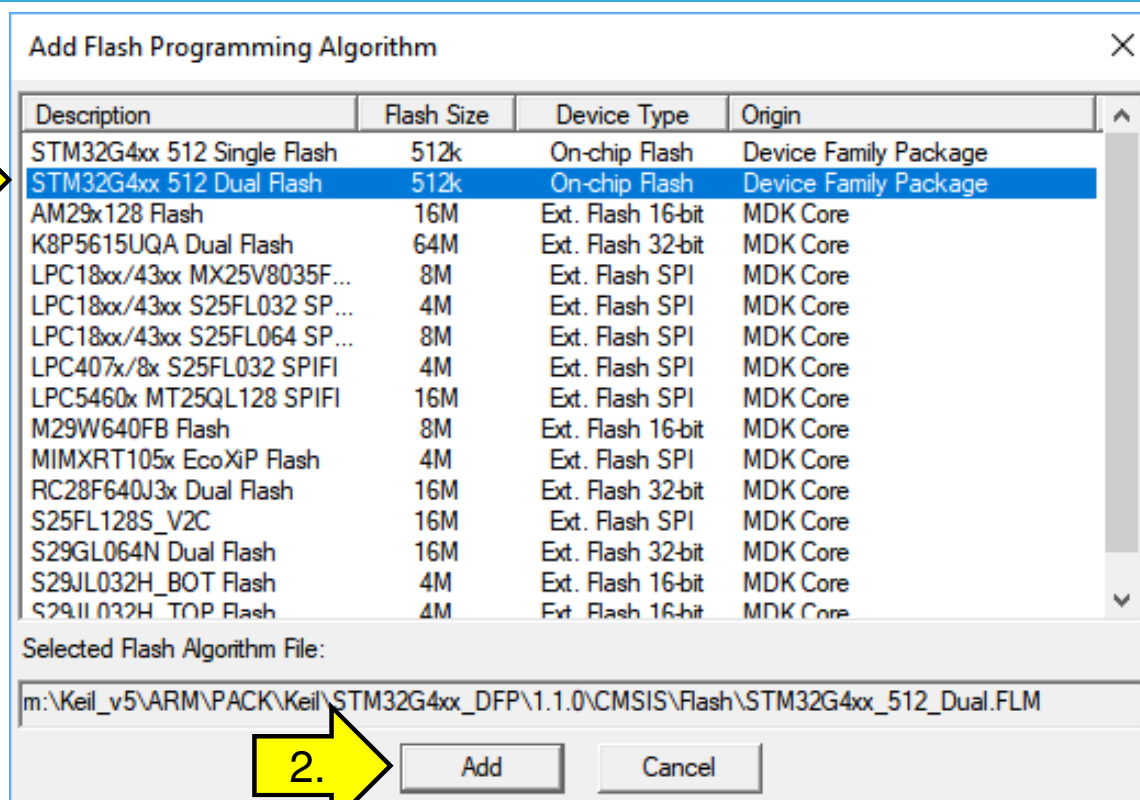


Flash Download → Add

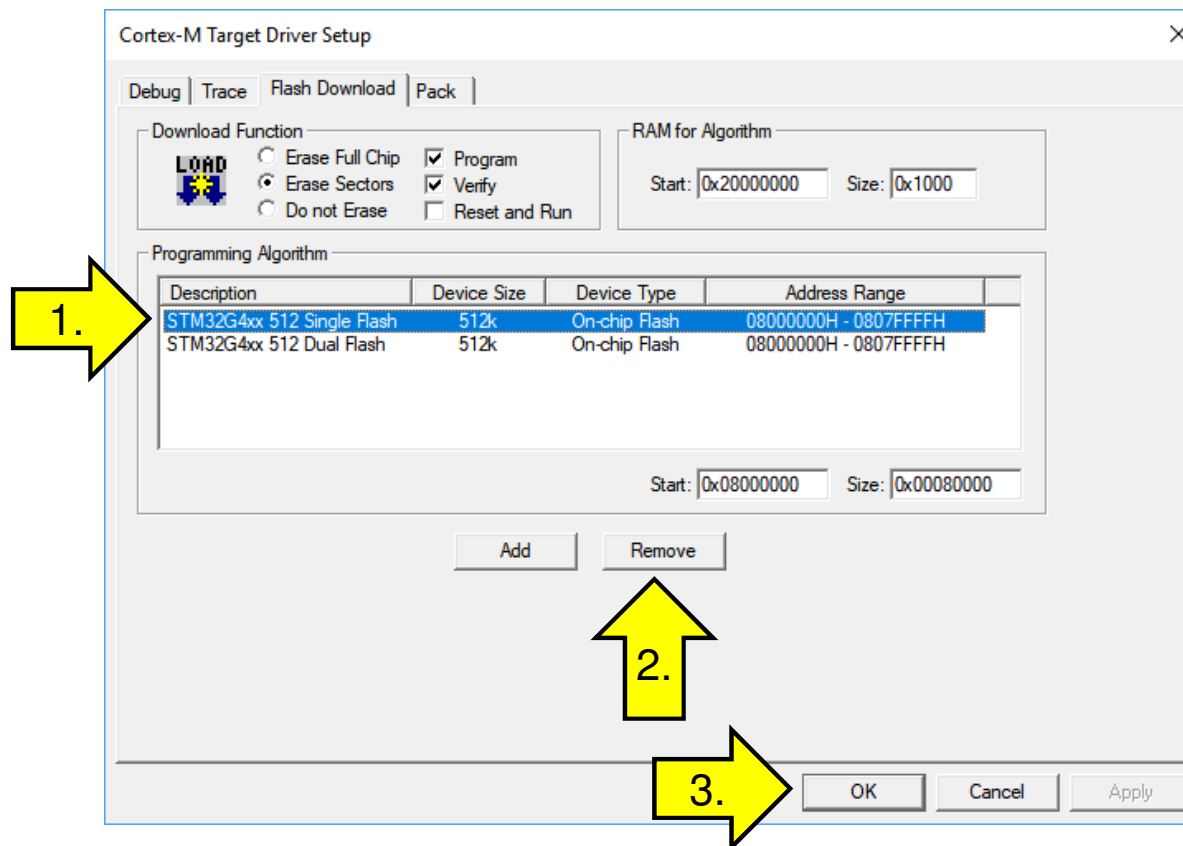




STM32G4xx 512 Dual Flash

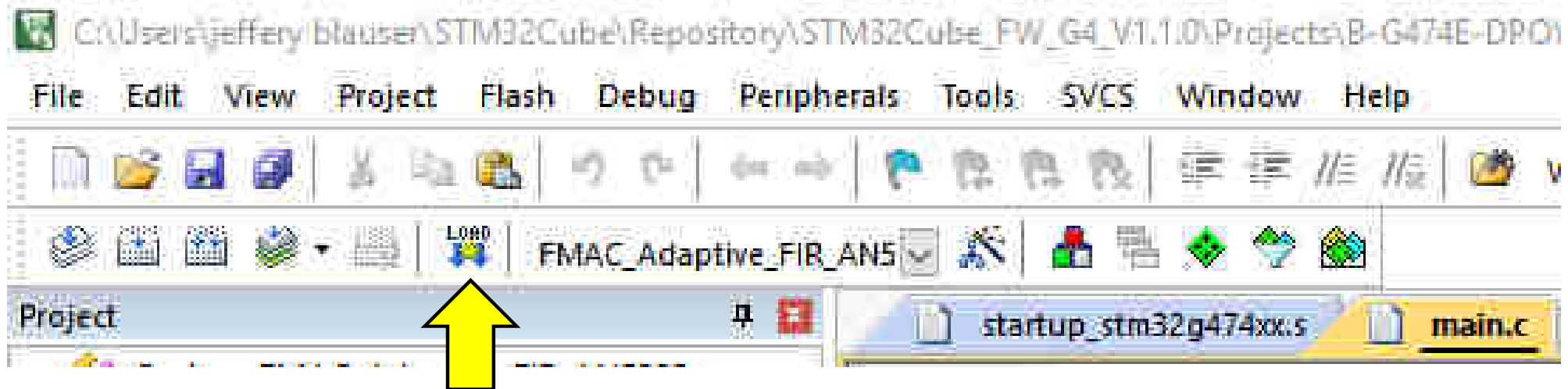


Remove Single Flash





Load



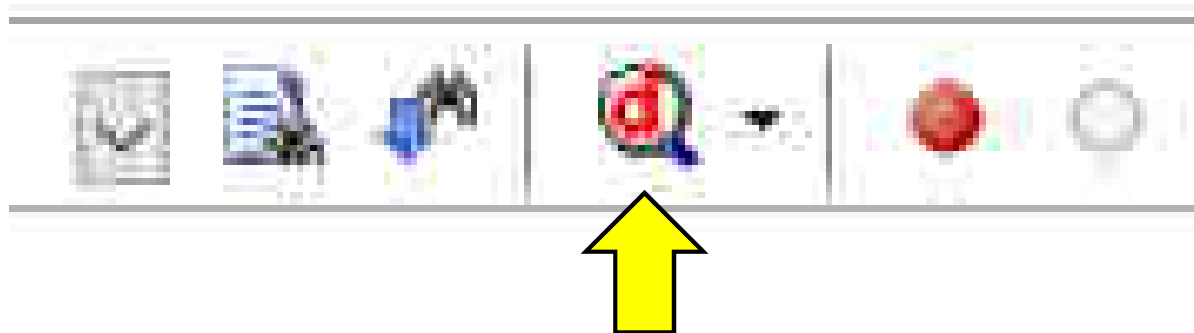
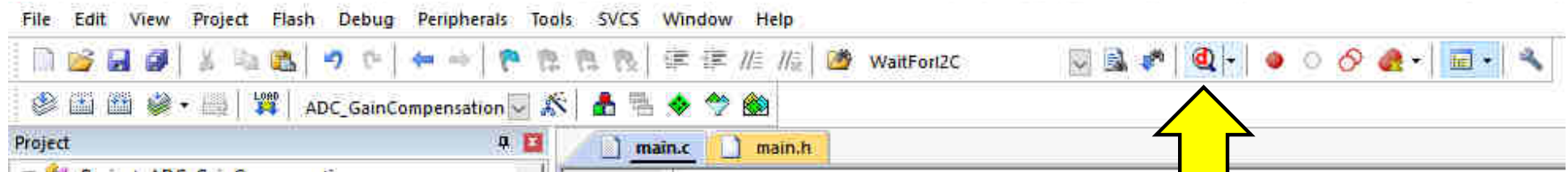
```
Build Output
Build Time Elapsed: 00:01:00
Load "FMAC_Adaptive_FIR_AN5305\\FMAC_Adaptive_FIR_AN5305.axf"
Erase Done.
Programming Done.
Verify OK.
Flash Load finished at 10:49:38
```



Keil MDK-ARM

327

Debugger

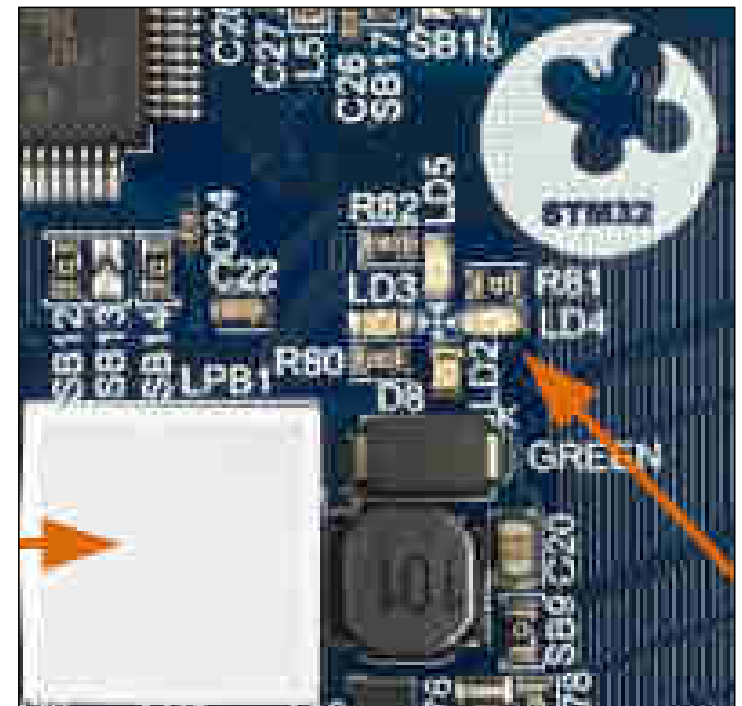
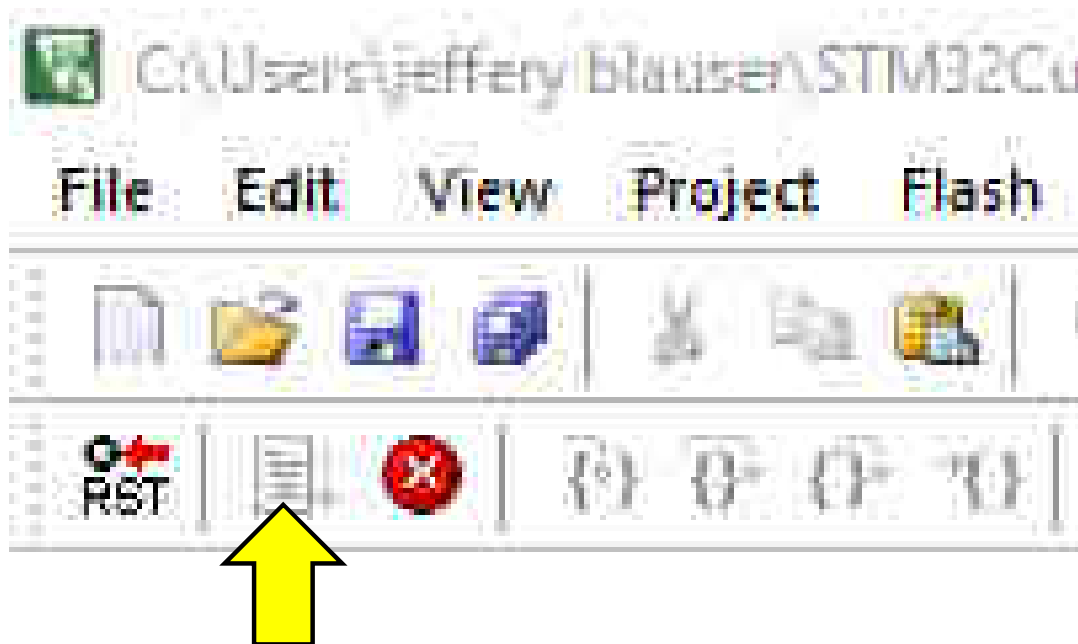




Keil MDK-ARM

328

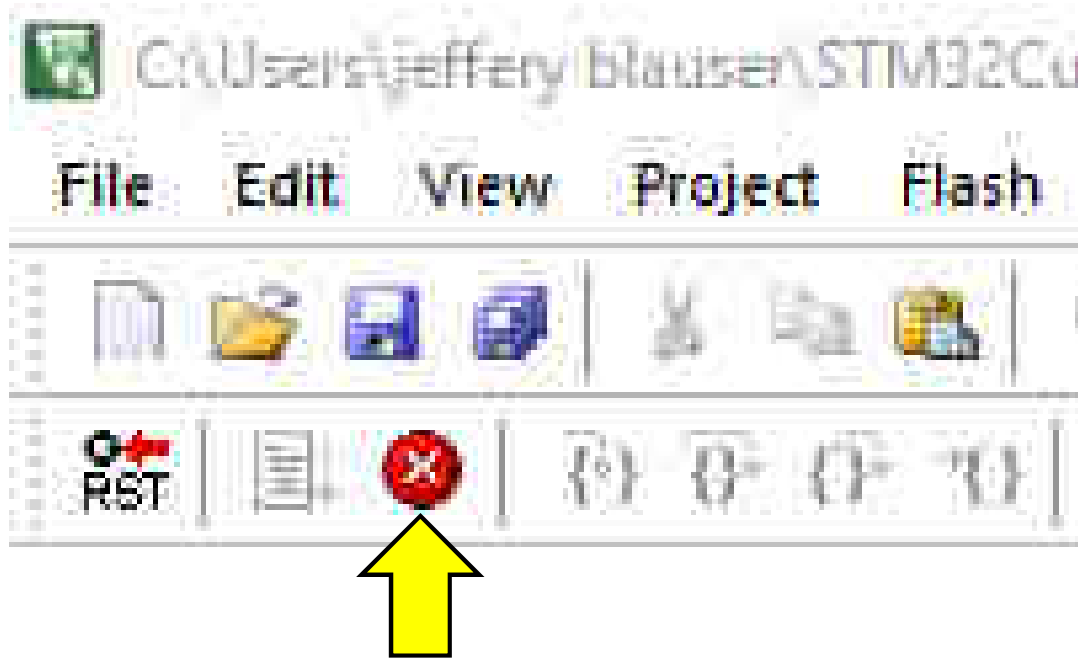
RUN



LD4 Illuminates



STOP Execution



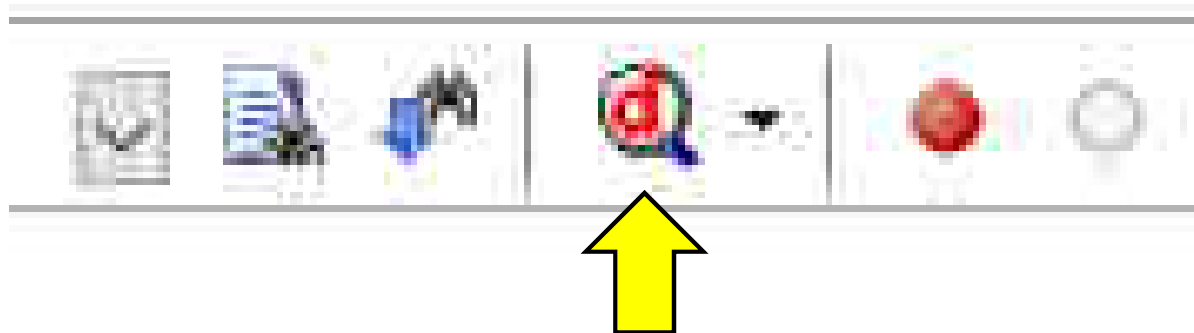
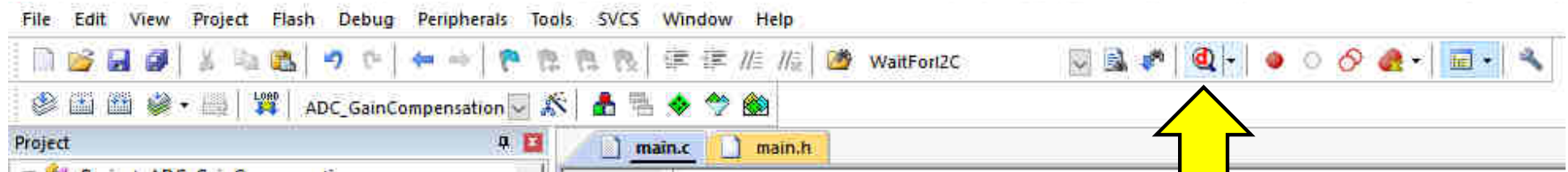
Beyond lighting an indicator LED, let's make it more interesting



Keil MDK-ARM

330

Debugger STOP to re-open the Editor





Filter Math Accelerator

331

Recall - Lab steps 6 of 6

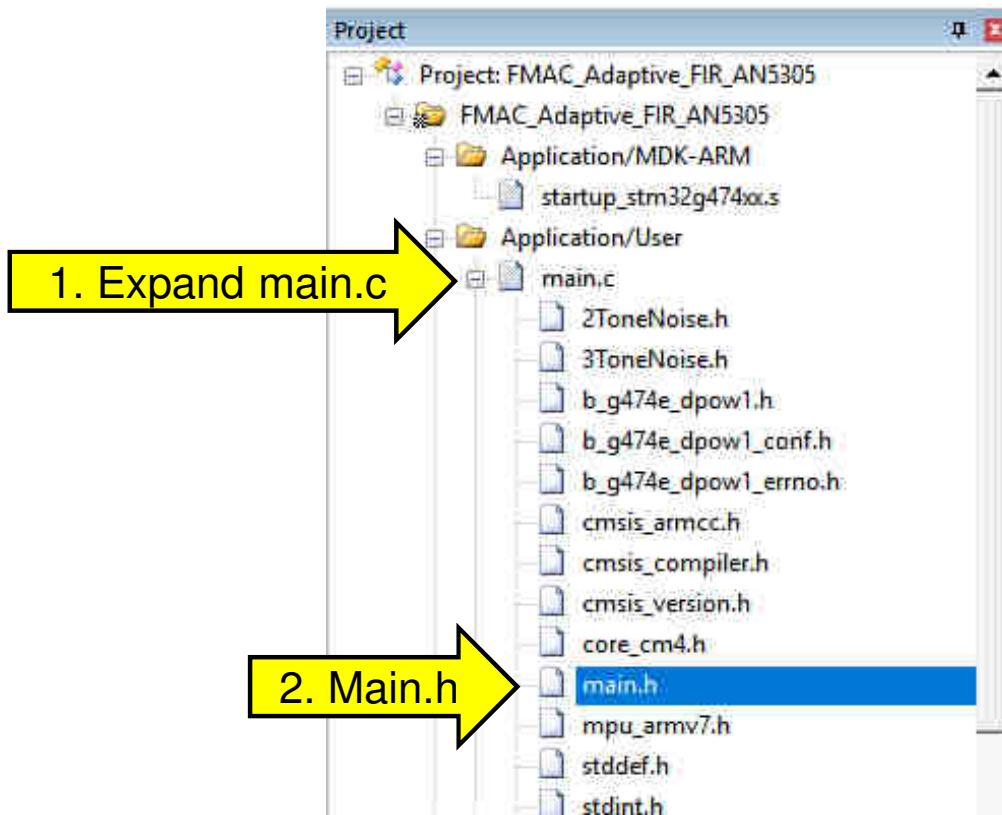
STM32 board LEDs are used to monitor the example status:

- LED4 is ON when correct FMAC FIR results have been calculated.
- LED5 is blinking (1 second period) if an error is detected in the FIR filter results or if there is an initialization or configuration error.

The filtered output is displayed in the Terminal I/O if the compilation switch
PRINT_OUTPUT is enabled in main.h.

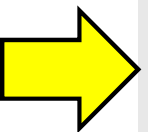


Finding main.h





main.h about line 119

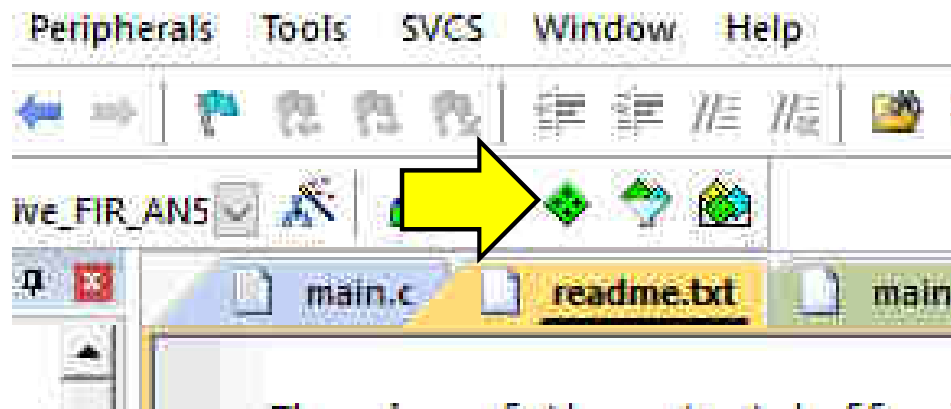


```
112  /* FMAC internal memory configuration: base address of the output buffer */
113  #define OUTPUT_BUFFER_BASE      COEFFICIENT_BUFFER_SIZE + INPUT_BUFFER_SIZE
114
115  /* FMAC internal memory configuration: size of the input buffer */
116  #define OUTPUT_BUFFER_SIZE      MEMORY_PARAMETER_D2
117
118  /* Print results to standard IO */
119  #define PRINT_OUTPUT
120
121  /* USER CODE END Private defines */
122
```

Confirm **#define PRINT_OUTPUT** is not commented out



Check the Run Time Environment (green diamond)



Is stdio enabled to support printf?



Compiler – I/O – STDOUT – Check box - ITM

Manage Run-Time Environment

Software Component

- Board Support
- CMSIS
- CMSIS Driver
- CMSIS Driver Validation
- CMSIS RTOS Validation
- Compiler
 - Event Recorder
 - I/O
 - File
 - STDERR
 - STDIN
 - STDOUT**
 - TTY

Sel.	Variant	Version	Description
	STM32G474E-EVAL	1.0.0	STMicroelectronics STM32G474E-EVAL Board
			Cortex Microcontroller Software Interface Components
			Unified Device Drivers compliant to CMSIS-Driver Specifications
	API	1.0.0	Run API test for enabled drivers
			CMSIS-RTOS Validation Suite
	ARM Compiler	1.6.0	Compiler Extensions for ARM Compiler 5 and ARM Compiler 6
☐	DAP	1.4.0	Event Recording and Component Viewer via Debug Access Port (DAP)
			Retarget Input/Output
☐	File System	1.2.0	Use retargeting together with the File System component
☐	Breakpoint	1.2.0	Stop program execution at a breakpoint when using STDERR
☐	Breakpoint	1.2.0	Stop program execution at a breakpoint when using STDIN
☒	ITM	1.2.0	Redirect STDOUT to a debug output window using ITM
☐	Breakpoint	1.2.0	Stop program execution at a breakpoint when using TTY

1. Arrow pointing to Compiler

2. Arrow pointing to I/O

3. Arrow pointing to STDOUT

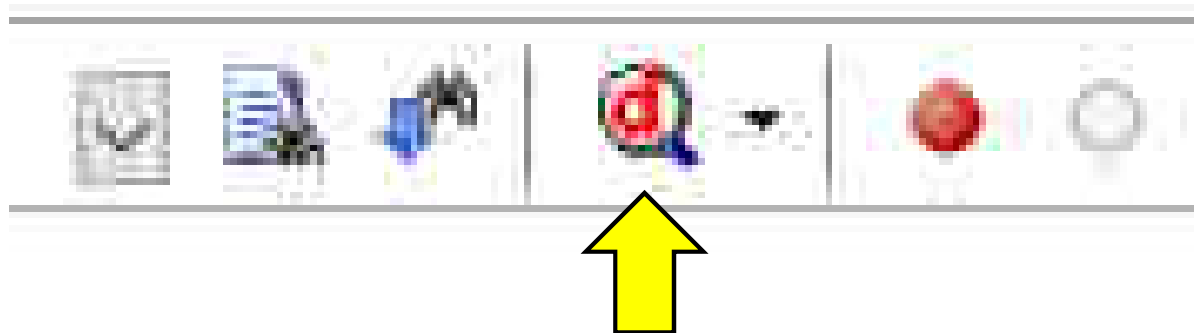
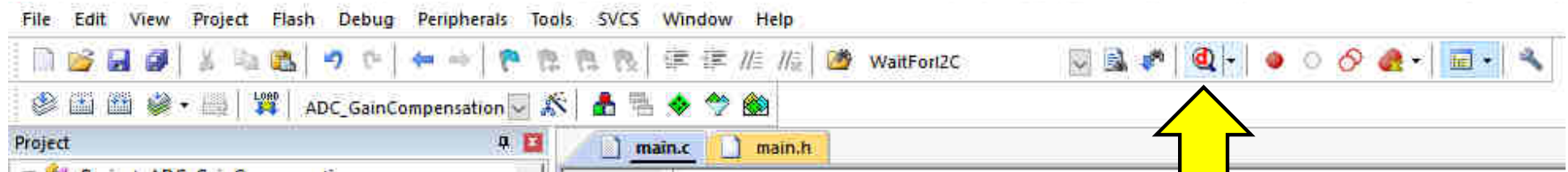
4. Arrow pointing to ITM



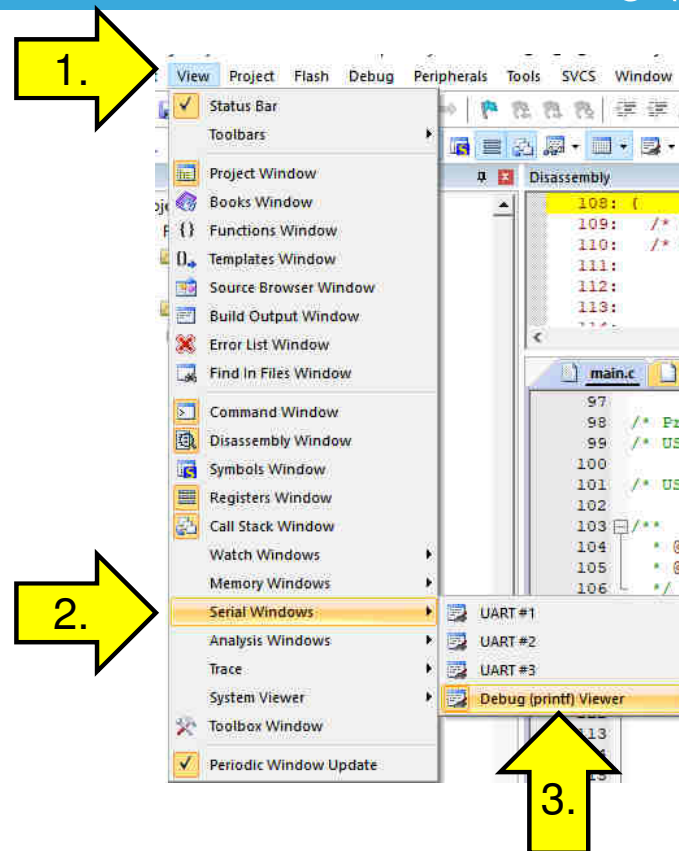
Keil MDK-ARM

336

Debugger

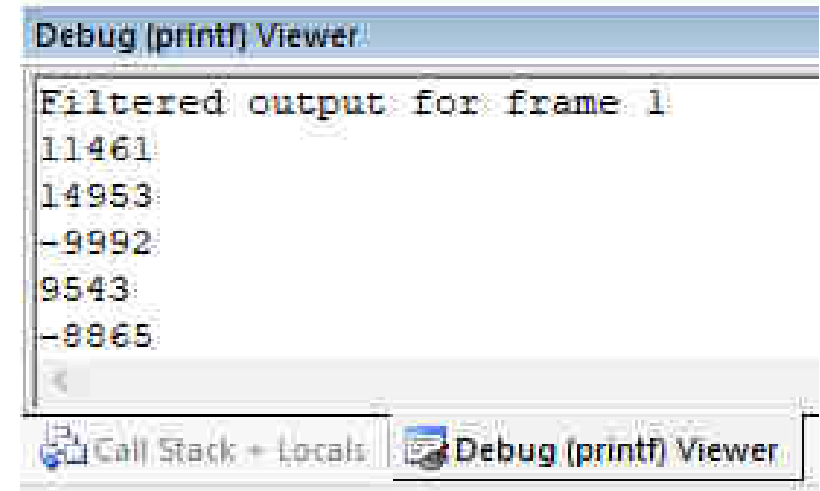
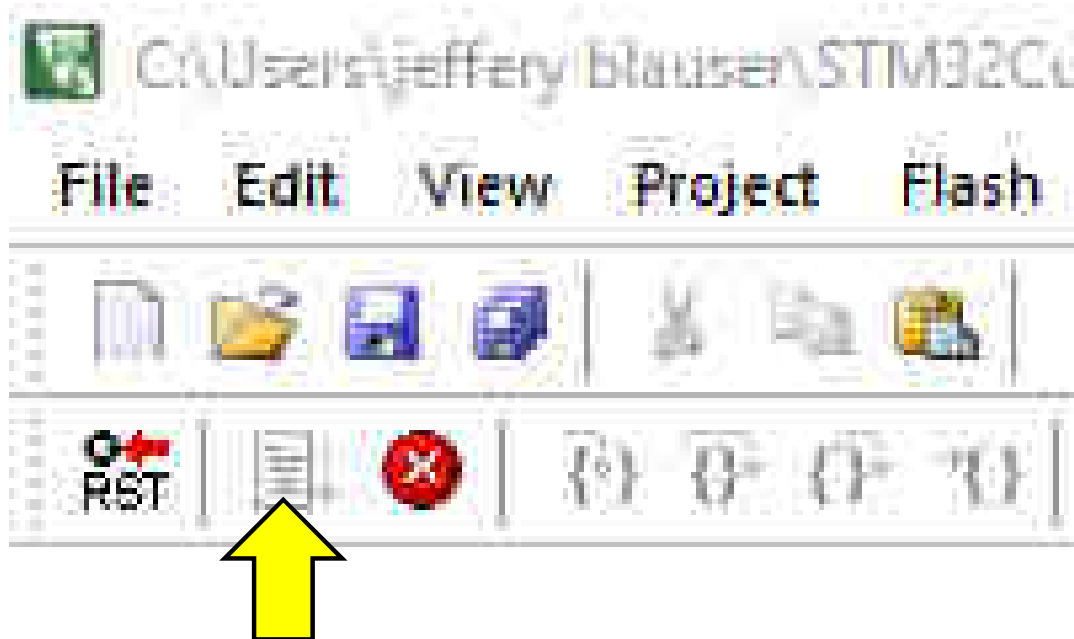


View – Serial Windows – Debug (printf) Viewer





RUN – Values in the Debug (printf) Viewer





Filter Math Accelerator

339

Recal - Lab steps 1 of 6

This Hands On Lab illustrates noise cancellation using an adaptive FIR filter.

The experiment will be run on two input frames:

- a gaussian noise with two tone interferers.
- a gaussian noise with three tone interferers.

Create individual coefficient sets.

The result is a "whitened" noise shown here.



Figure 3. 'Whitened' noise (frame 1)

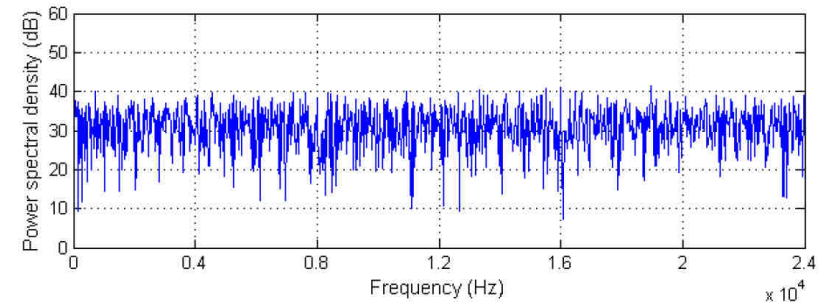
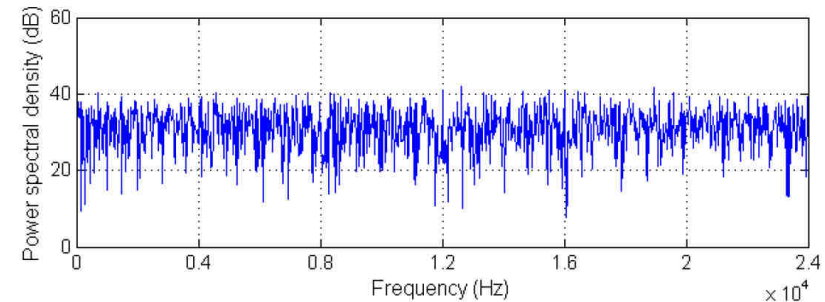
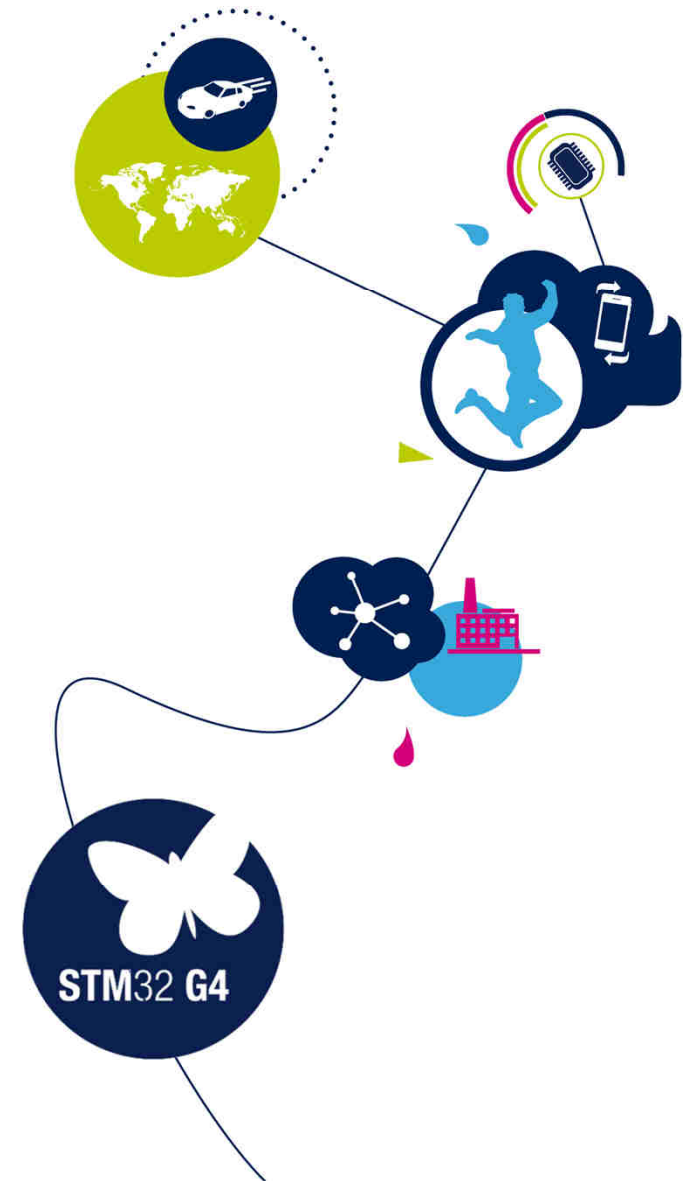


Figure 6. 'Whitened' noise (frame 2)



STM32G4 mixed-signal MCU workshop

*Hands On:
Dual Bank Flash
Live Firmware Upgrade*





Dual Bank flash

341

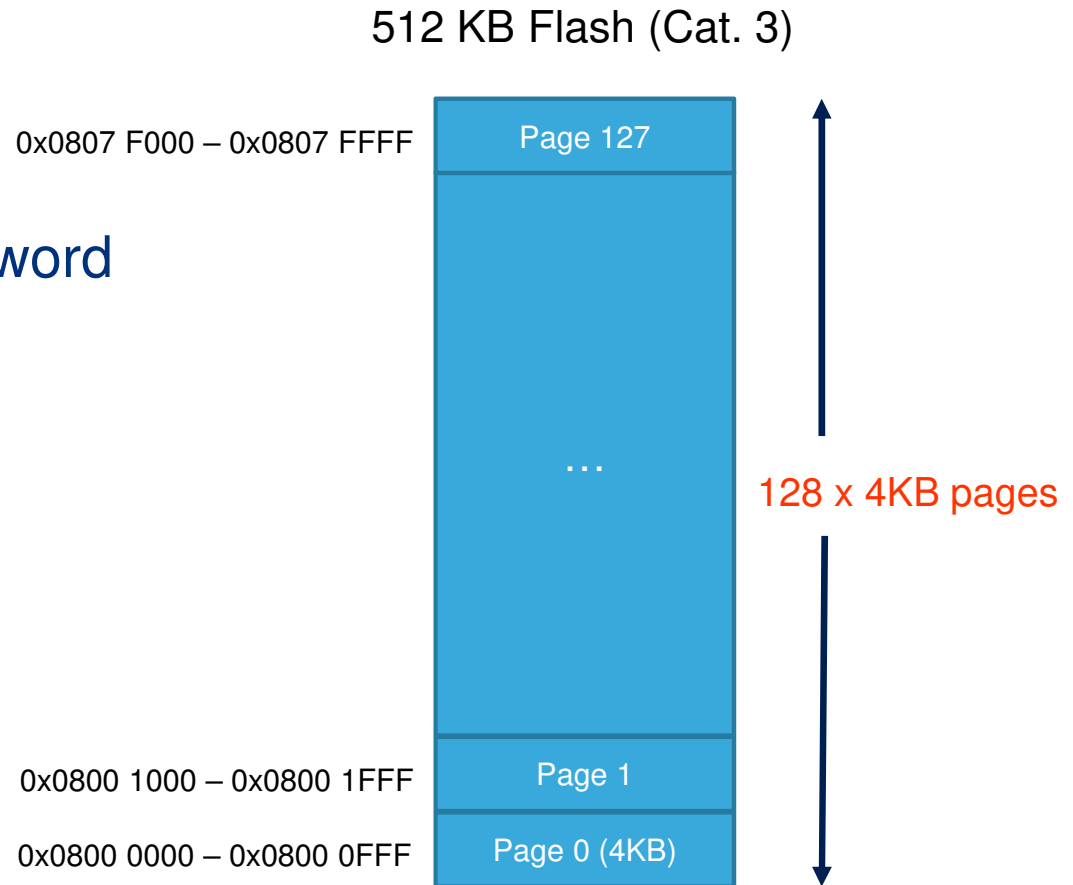
Objective:

- To show the application running from one bank and writing to the second bank(Read-While-Write). Demonstrate the live upgrade possibility.
- Using a Terminal (TeraTerm) to send updated application binary file via the Virtual Comm Port.

512 KB die Flash Organization

342

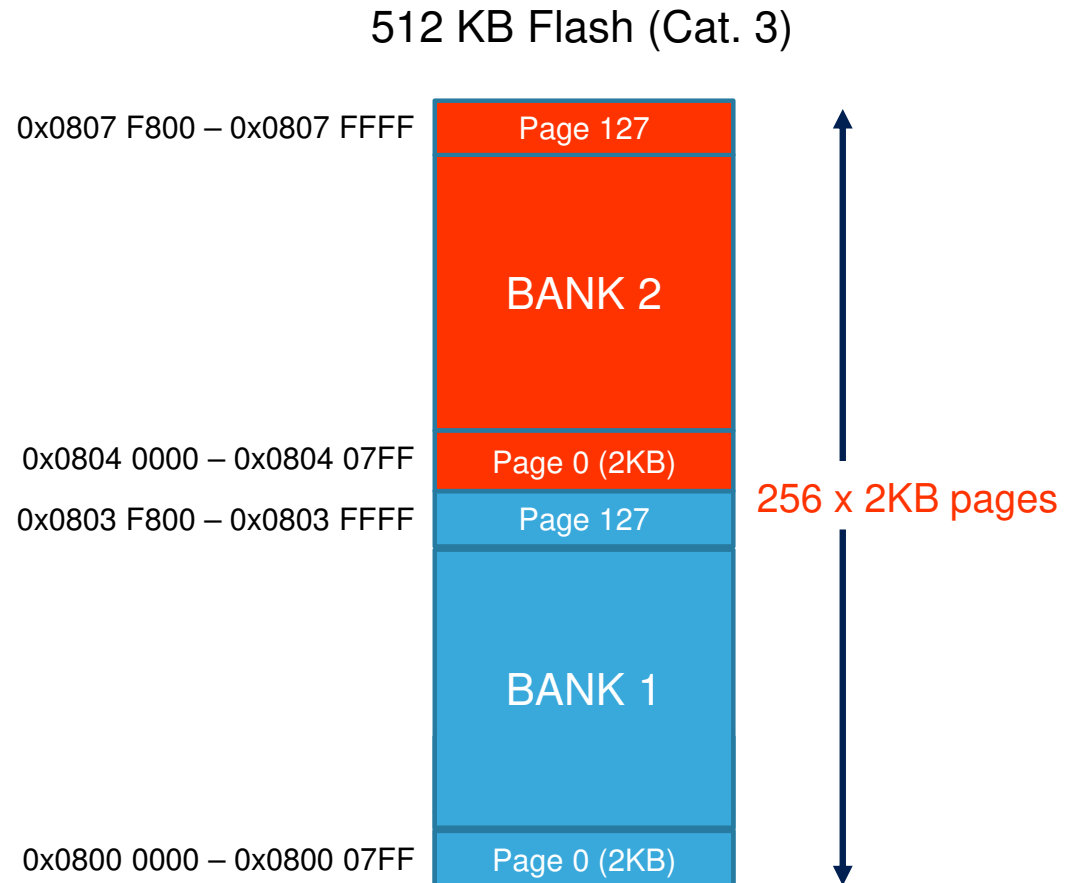
- DBANK = 0 (Single Bank)
 - 128 bits read access
- ECC: 8 bits per 64 bit double-word
- 4 Write protection areas
- Page and Mass erase



512 KB die Flash Organization

343

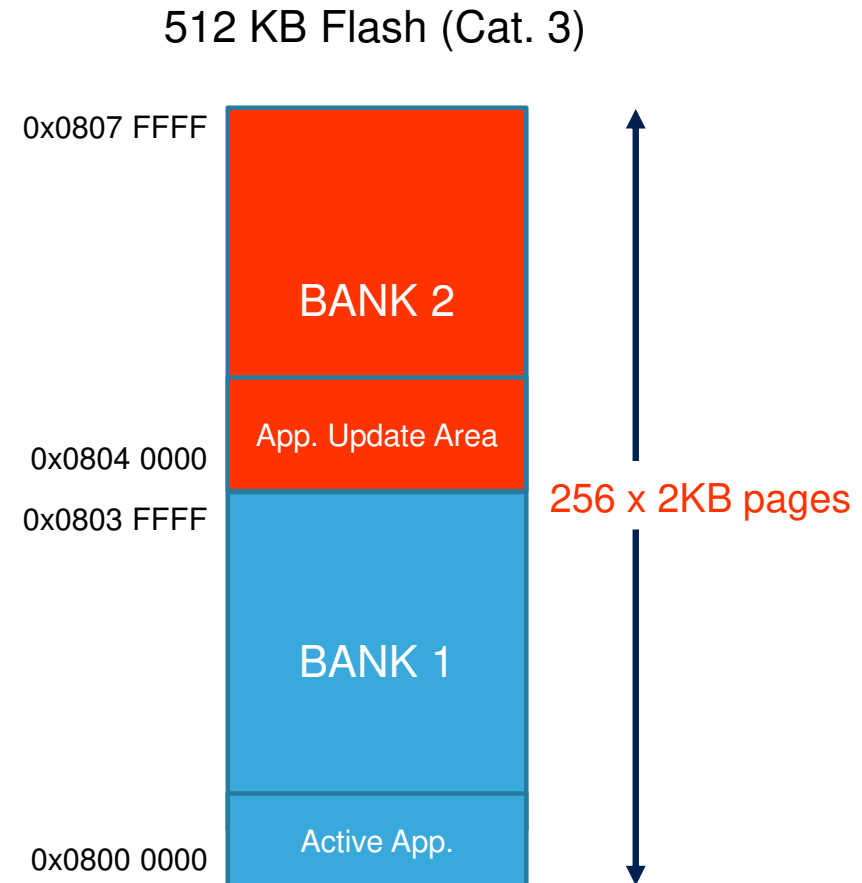
- DBANK = 1 (Dual Bank - Default)
 - 64 bits read access
- Read-While-Write (RWW)
- ECC: 8 bits per 64 bit double-word
- 2 Write protection areas per bank
- Page, Bank and Mass erase



512 KB die Flash Organization

344

- Active application executing from Bank 1
- Using serial port to receive update
 - Temporary storage in RAM
- Write to BANK 2 area
- Switch application execution to BANK 2





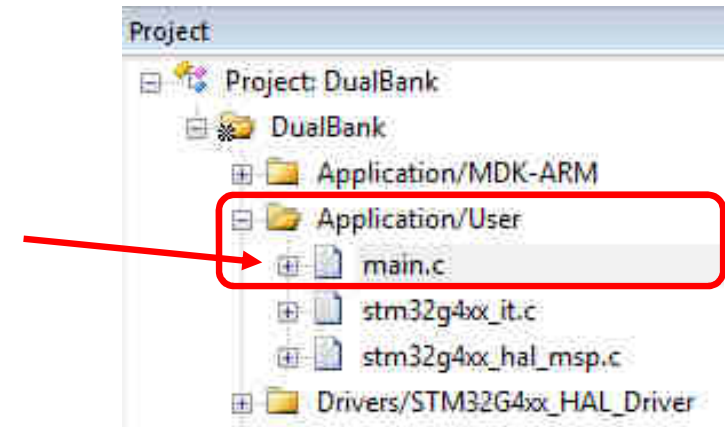
Hands-On: Dual Bank flash

345

- Open the DualBank project in Keil MDK-ARM folder
 - DualBank\MDK-ARM\DualBank.uvprojx



- Expand “Application/User” folder
- Open “main.c” file
 - Initialize the Clocks, GPIO and Peripherals
 - Main while loop (State Machine)
 - Handles BANK 2 page erase
 - Handles BANK 2 page write
 - Jump and Execute in Bank 2





- Timer 2 IRQ Callback(Line 481)
 - 10 Hz period
 - LD_ptr: Tracks LED to toggle
- GPIO EXTI Callback (Line 509)
 - Detects rising edge of JOYSTICK_SEL button
- UART Rx complete Callback (Line 524)
 - Signals that the USART Rx DMA has completed

IRQ Callback functions

346

```
481 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
482 {
483     switch ( LD_ptr )
484     {
```

```
509 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
510 {
511     if ( ( GPIO_Pin == JOYSTICK_SEL_Pin ) && ( UpdateState == 0 ) )
```

```
524 void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
525 {
526     rx_complete = 1;
527 }
```



Peripheral configuration

347

- **JOYSTICK_SEL**: EXTI13
 - PC13
- **LD2-5**: GPIO_Output
 - PA15, PB1, PB5 and PB7
- **ST-LINK VCP**: USART3
 - PC10 and PC13
- **Serial Wire Debug with Trace (SWO)**:
 - PA13, PA14 and PB3

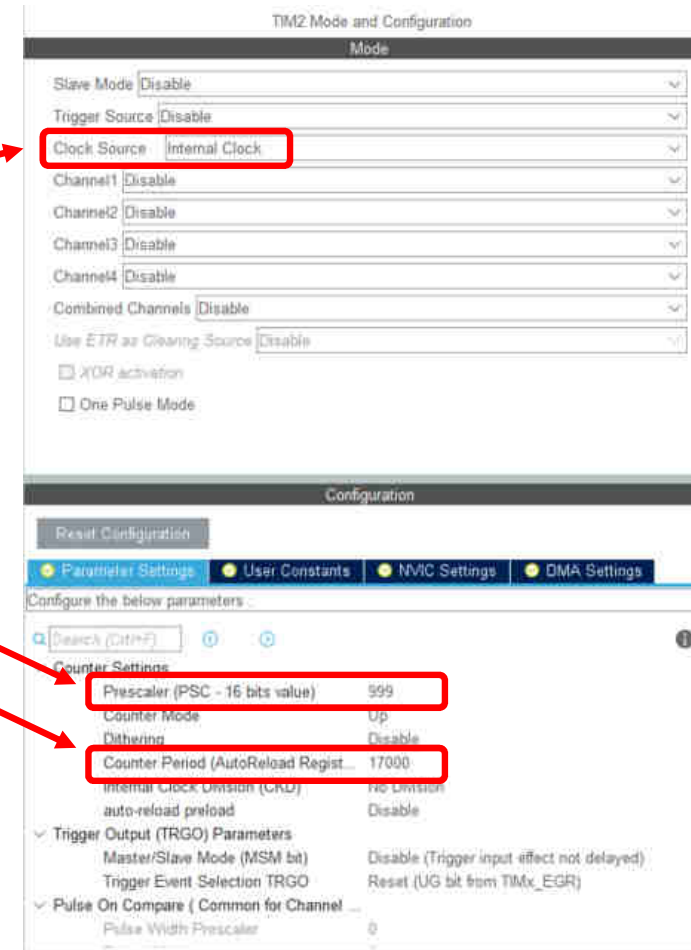
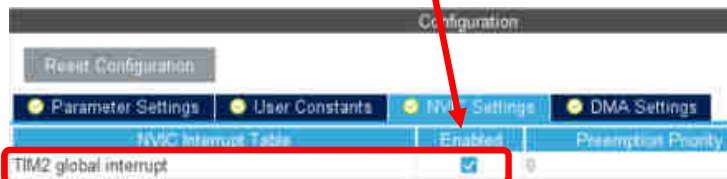




Peripheral configuration

348

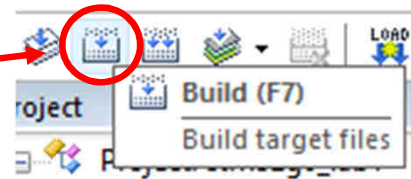
- **Timer 2:** LED blink rate timer
- Internal Clock Source = 170 MHz
- Prescaler = 999
- Counter Period = 17000
 - 10 Hz timebase interrupt
- TIM2 global interrupt Enabled



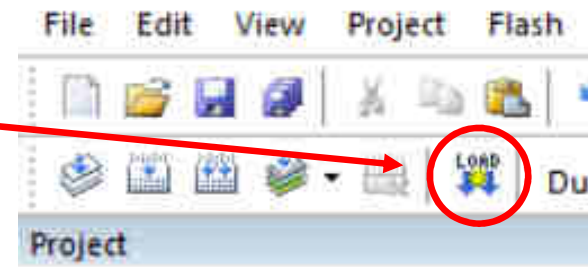
Build the Project

349

- “Build” project



- “LOAD” project to flash



- Press the “RESET” button on the Discovery
 - 4 LEDs blinking in rotating pattern.

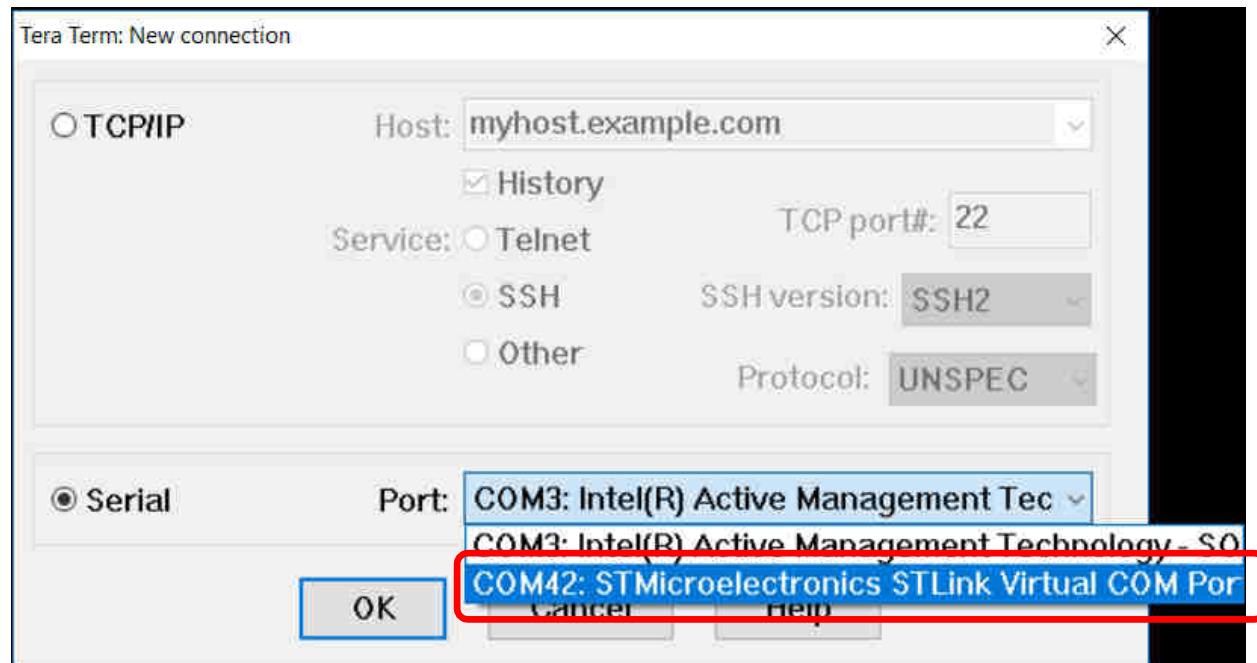
Hands On: Dual Bank

350

- Launch “TeraTerm”



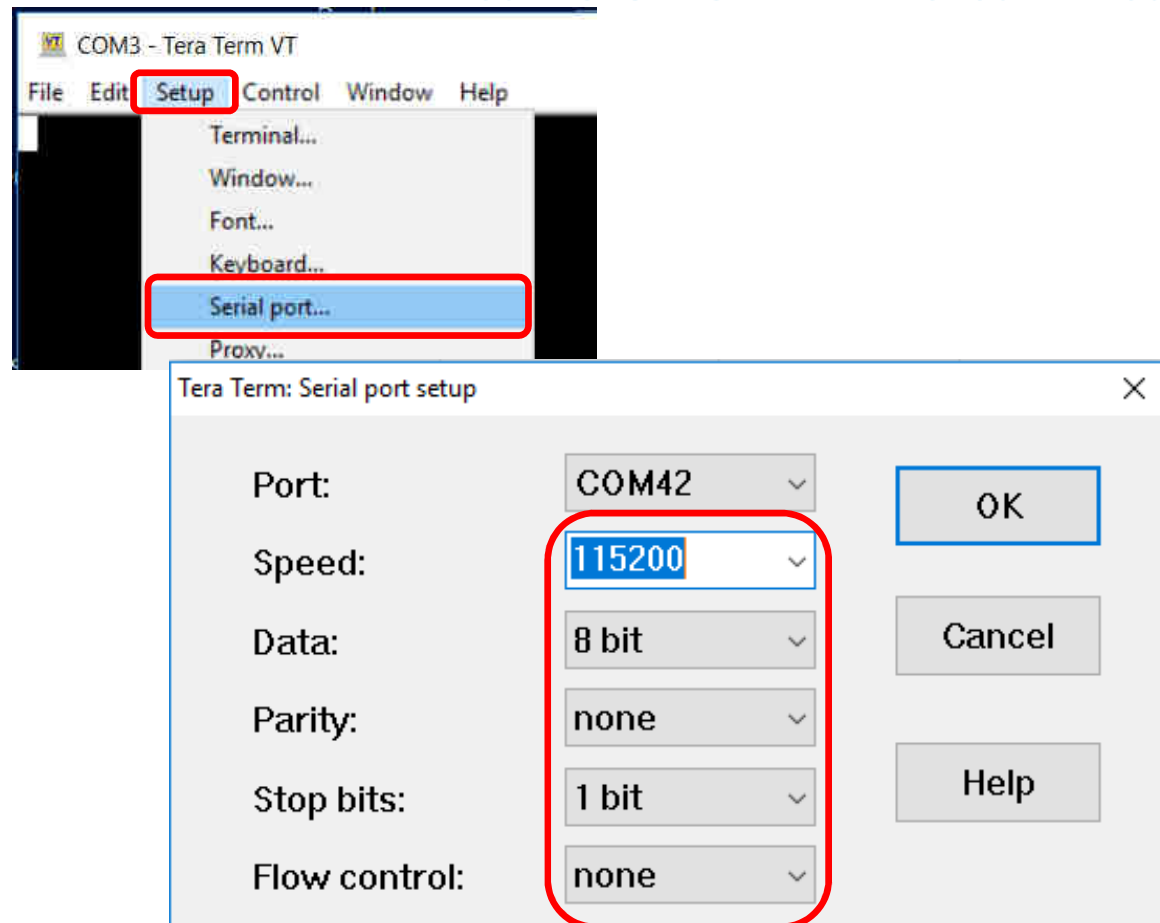
- Open “New Connection”
 - Click “Serial” button
 - Change Port to “STLink Virtual COM Port”
 - Click “OK”



Hands On: Dual Bank

351

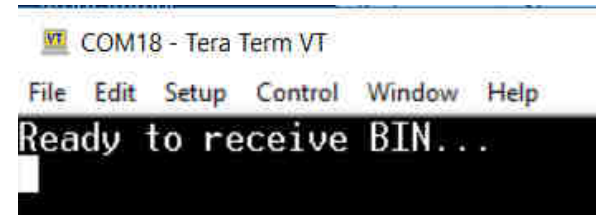
- Verify “Serial port..”
 - Baud = 115200
 - Data = 8 bit
 - Parity = none
 - Stop = 1 bit
 - Flow control = none



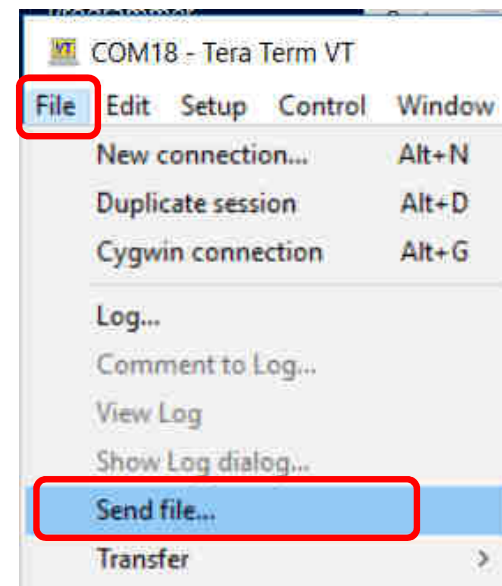
Hands On: Dual Bank

352

- Press “JOYSTICK_SEL” on Discovery



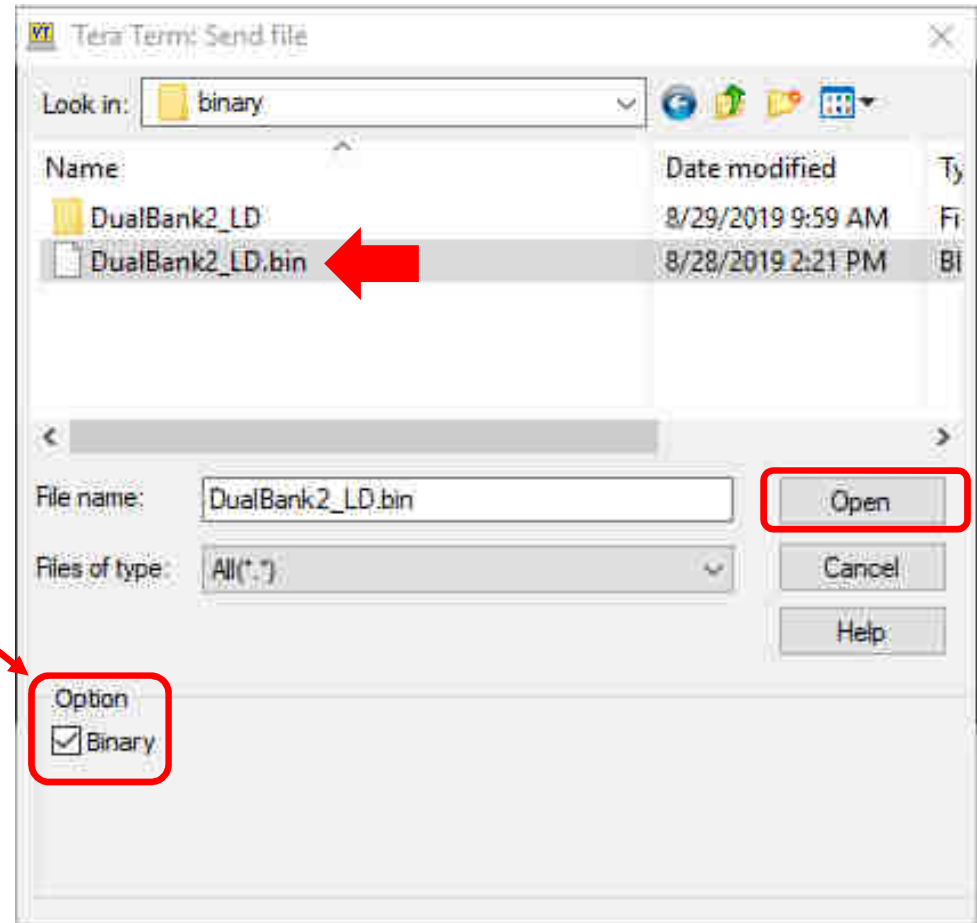
- Send binary file in TeraTerm
 - File > Send file...



Hands On: Dual Bank

353

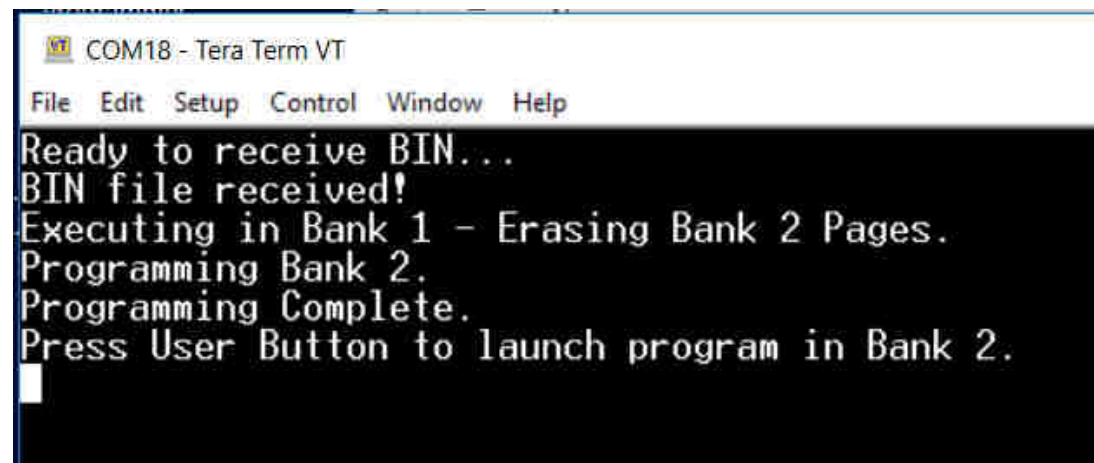
- Select “DualBank2_LD.bin” file
 - DualBank\binary folder
- Check “Binary” Option
- Click “Open”



Hands On: Dual Bank

354

- TeraTerm will display programming status



A screenshot of a TeraTerm VT window titled 'COM18 - Tera Term VT'. The window has a menu bar with 'File', 'Edit', 'Setup', 'Control', 'Window', and 'Help'. The main text area shows the following status messages: 'Ready to receive BIN...', 'BIN file received!', 'Executing in Bank 1 - Erasing Bank 2 Pages.', 'Programming Bank 2.', 'Programming Complete.', and 'Press User Button to launch program in Bank 2.'. A small white cursor is visible at the end of the last line.

```
COM18 - Tera Term VT
File Edit Setup Control Window Help
Ready to receive BIN...
BIN file received!
Executing in Bank 1 - Erasing Bank 2 Pages.
Programming Bank 2.
Programming Complete.
Press User Button to launch program in Bank 2.
█
```

- Press “User Button” to launch program in Bank 2
 - LEDs blink pattern has changed.



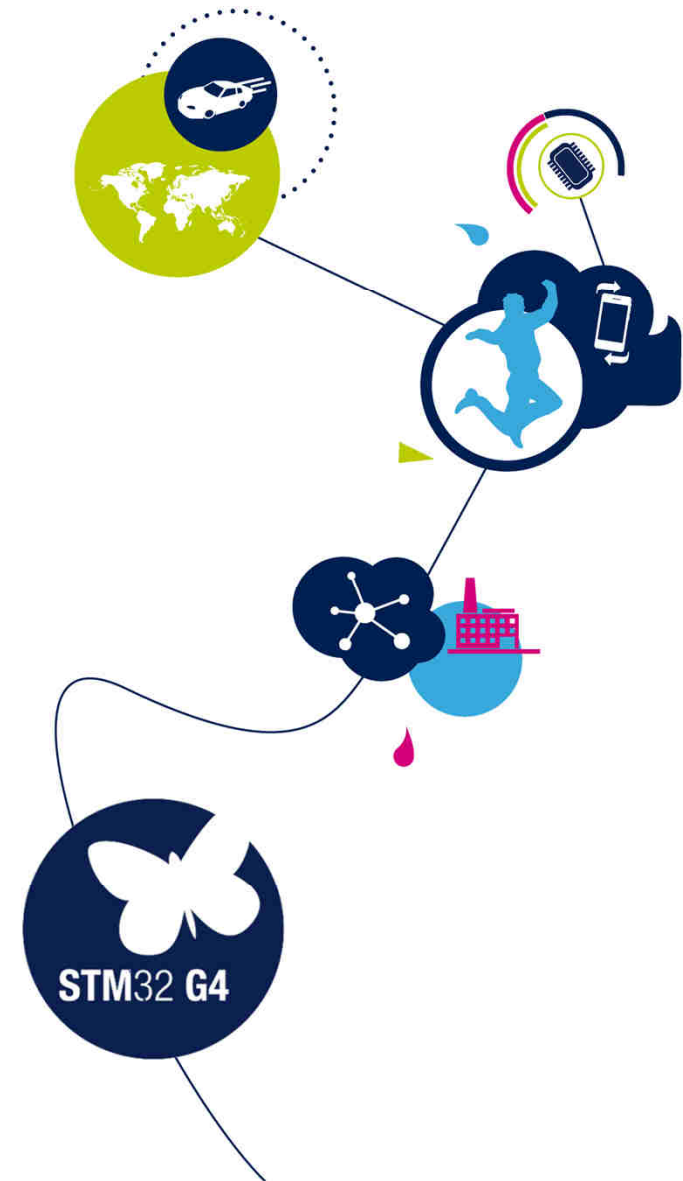
Dual Bank: Additional Notes

355

- Push board “RESET” button to switch execution back to Bank 1
 - To select boot from Bank 2, set the BFB2 bit in the user option bytes
- Reference Manual RM0440: STM32G4 Series advanced ARM-based 32-bit MCUs
 - Section 3: Embedded Flash memory for category 3 devices
- AN2606: STM32 microcontroller system memory boot mode

STM32G4 mixed-signal MCU workshop

Hands On: CCM SRAM





Hands-On: CCM SRAM

357

Objective:

1. Illustrate how to place functions in CCM SRAM.
2. Illustrate how to place files in CCM SRAM.
3. Illustrate how to place library modules in CCM SRAM.



CCM SRAM Advantages

358

- Tightly coupled to ARM Cortex core
- Zero wait state with deterministic performance
- Decrease execution time for critical task
- Minimal Interrupt routines latency when placed in CCM SRAM



CCM SRAM (functions)

359

- Hands-On: Place function in CCM SRAM

- Open the CCM1.ioc CubeMx project (double mouse click)

- Labs\CCM\CCM1\CCM1.ioc

MX CCM1.ioc

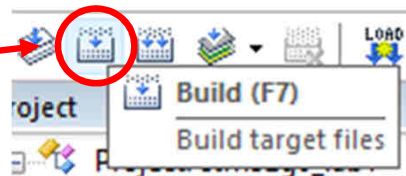
- Generate Code



- Open Project



- Build Project

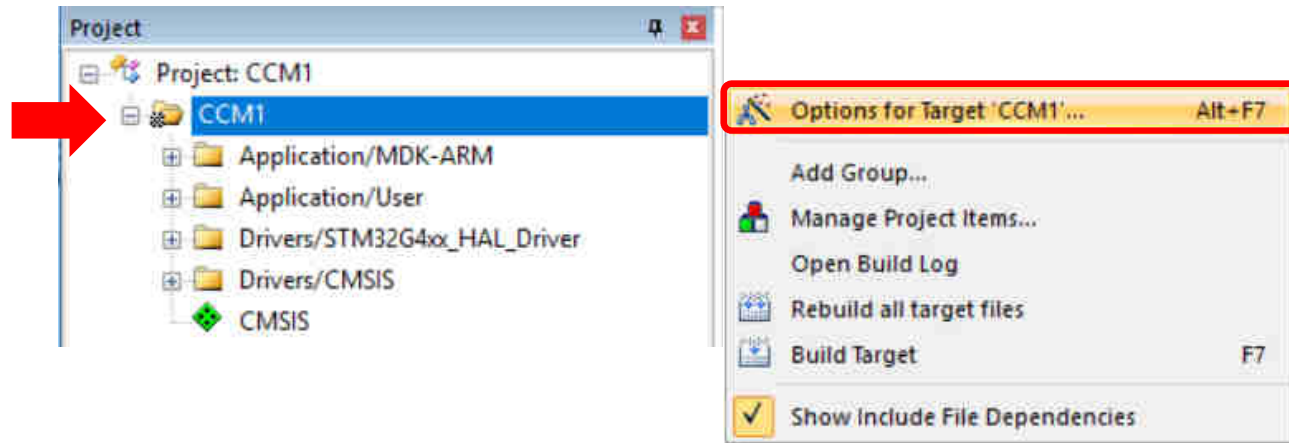




CCM SRAM (functions)

360

- Define a new memory region for the CCM SRAM in the MDK-ARM project scatter file
 - Right mouse click on CCM1
 - Select "Options for Target 'CCM1'..."

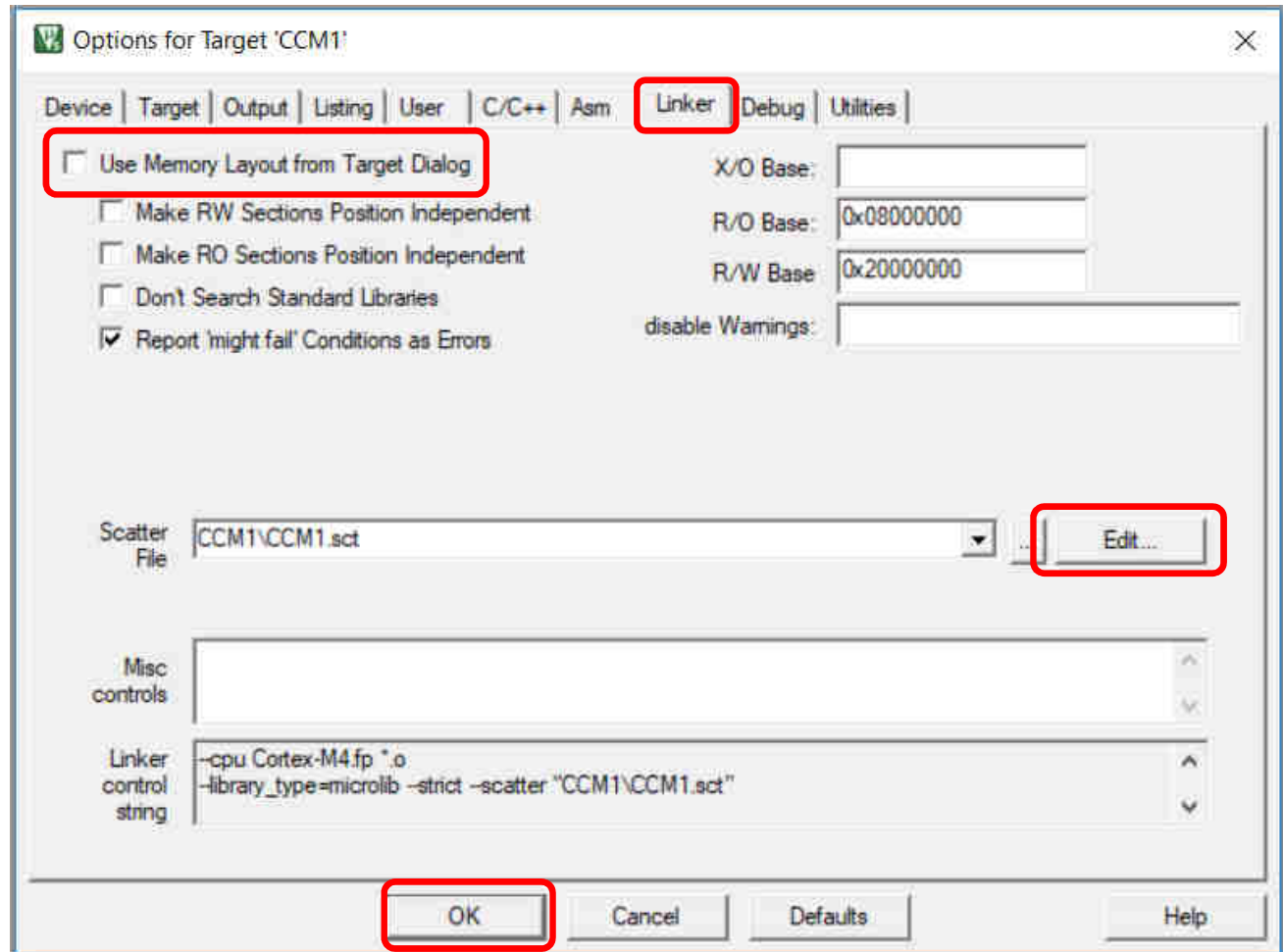




CCM SRAM (functions)

361

- Select “Linker” tab
- Uncheck “Use Memory Layout from Target Dialog”
- Select “Edit” to open scatter file
- Click “OK”





CCM SRAM (functions)

362

- Place a new line after line 14 (RW_IRAM1 region)
- Define a new CCMRAM region with the following:

```
ER 0x10000000 0x00002000 {  
  .ANY (ccmram)  
}
```

- Save CCM1.sct file

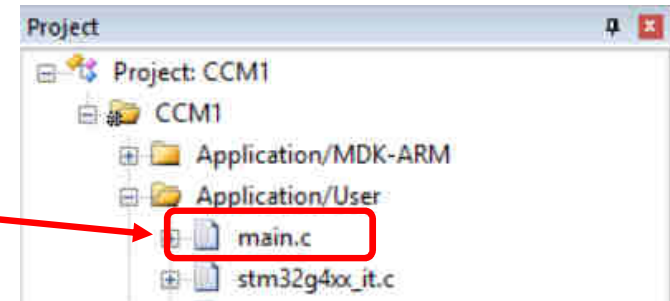
```
CCM1.sct  
1 ; *****  
2 ; *** Scatter-Loading Description File generated by uVision ***  
3 ; *****  
4  
5 LR_IROM1 0x08000000 0x00080000 { ; load region size_region  
6   ER_IROM1 0x08000000 0x00080000 { ; load address = execution address  
7     *.o (RESET, +First)  
8     *(InRoot$$Sections)  
9     .ANY (+RO)  
10    .ANY (+XO)  
11  }  
12 RW_IRAM1 0x20000000 0x00020000 { ; RW data  
13   .ANY (+RW +ZI)  
14 }  
15 ER 0x10000000 0x00002000 { ; load address = execution address  
16   .ANY (ccmram)  
17 }  
18  
19
```



CCM SRAM (functions)

363

- Expand “Application/User” folder
- Open “main.c” file
 - Scroll down to line 211



- Place the MDK-ARM keyword

`__attribute__((section("ccmram")))`

```
210 /* USER CODE BEGIN 4 */  
211 __attribute__((section("ccmram")))  
212
```

- Place keyword above each function to execute from CCM SRAM



CCM SRAM (functions)

364

- Place function to execute from CCM SRAM after `__attribute__` keyword
 - Copy and paste “HAL_GPIO_EXTI_Callback” function from CCM lab code.txt to the start of line 214

```
210  /* USER CODE BEGIN 4 */
211  __attribute__((section ("ccmram")))
212
213
214  void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
215  {
216      if ( GPIO_Pin == JOYSTICK_SEL_Pin )
217      {
218          HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_RESET);
219          HAL_GPIO_WritePin(LD3_GPIO_Port, LD3_Pin, GPIO_PIN_RESET);
```

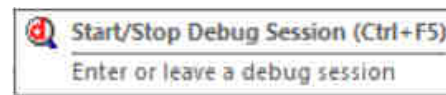
CCM SRAM (functions)

B65

- “Build” project



- “Start Debug” Session



- Place breakpoint in “HAL_GPIO_EXTI_Callback” (Line 216)

```
214 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
215 {
216     if ( GPIO_Pin == JOYSTICK_SEL_Pin )
217     {
```

- “Run” code



- Press Joystick button (Up, Down, Left, Right, Select)
- Breakpoint stops in CCM SRAM region address 0x10000018

```
222: }
0x10000018 F5B05F00 CMP r0,#0x2000
215: {
```



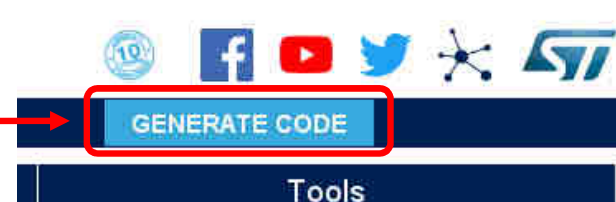
CCM SRAM (files)

366

- Hands-On: Place file(s) in CCM SRAM
- Open the CCM2.ioc CubeMx project (double mouse click)
 - Labs\CCM\CCM2\CCM2.ioc

MX CCM2.ioc

- Generate Code



- Open Project

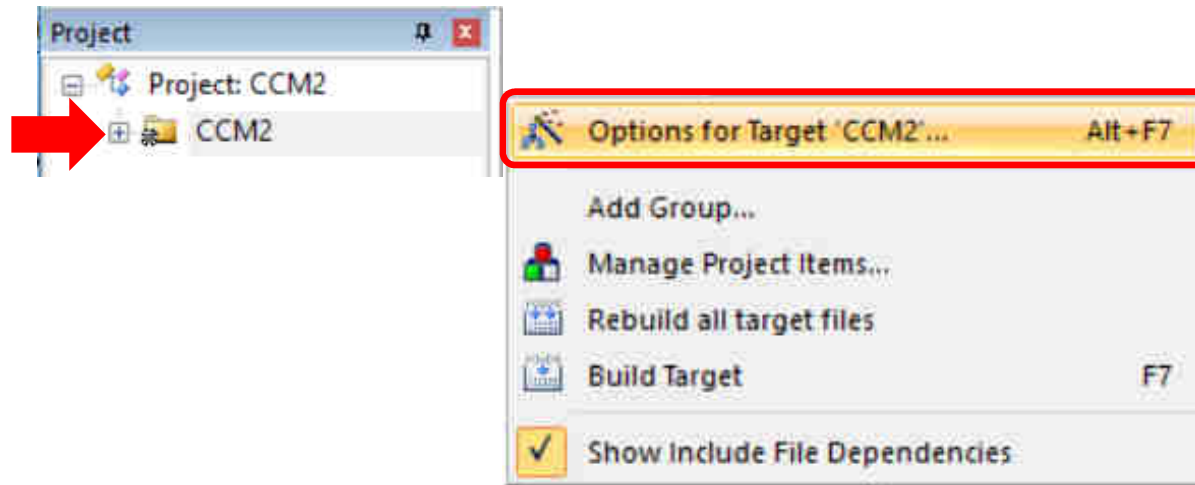




CCM SRAM (files)

367

- Define the memory region for the CCM SRAM in the MDK-ARM
 - Right mouse click on CCM2
 - Select "Options for Target 'CCM2'..."

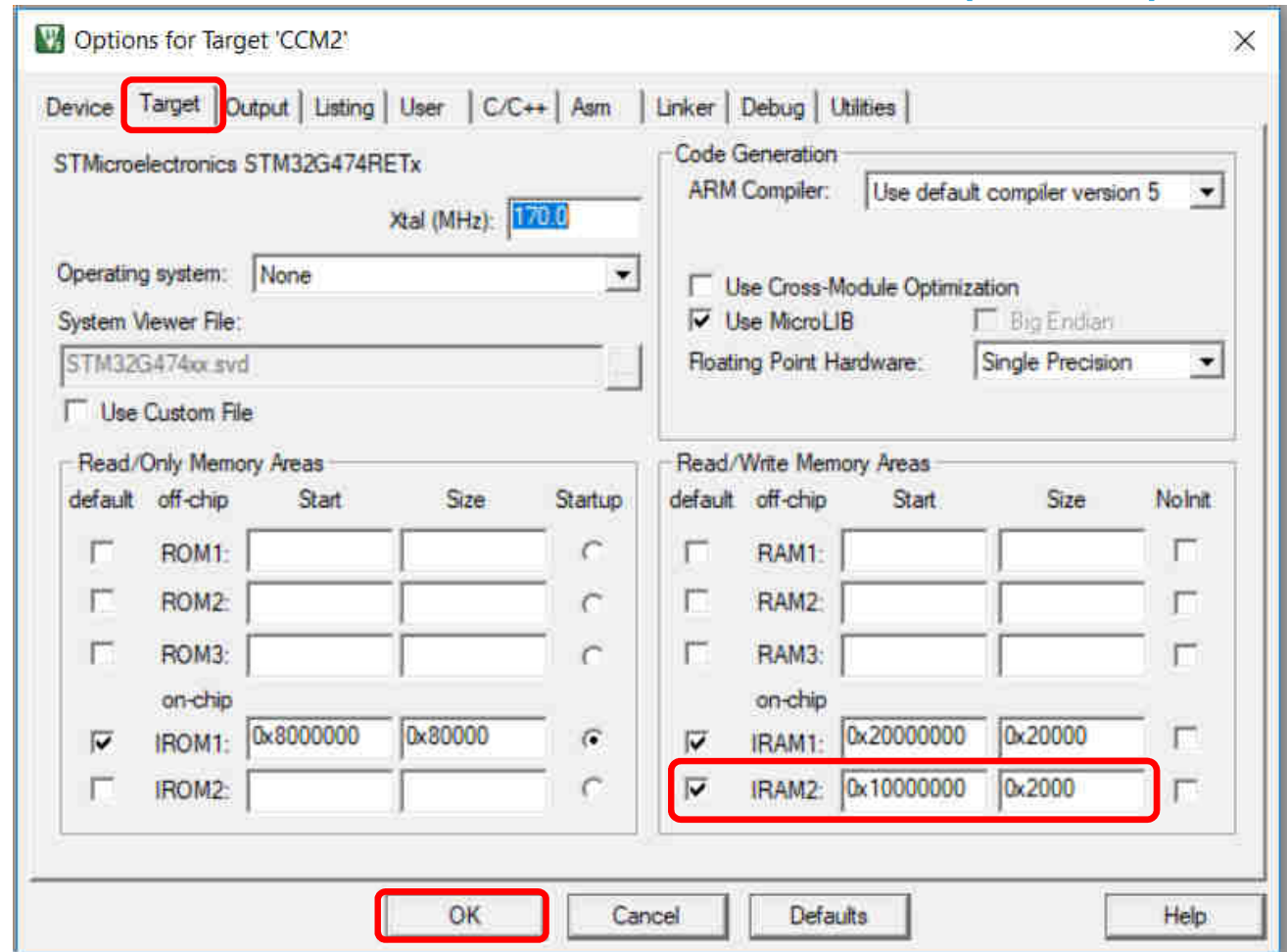




CCM SRAM (files)

368

- Select “Target” tab
- Check “IRAM2”
- Enter “0x10000000”
IRAM2 Start
- Enter “0x2000”
IRAM2 Size
- Click “OK”

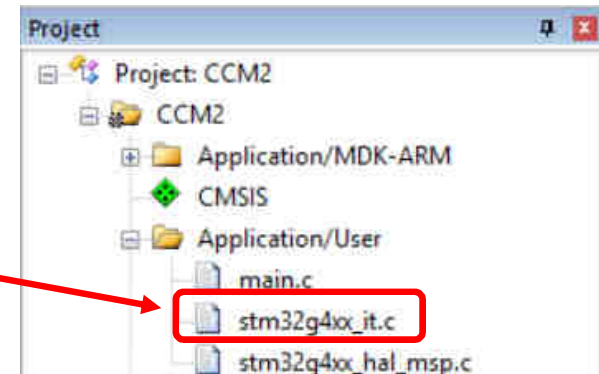




CCM SRAM (files)

369

- Expand “Application/User” folder
- Open “stm32g4xx_it.c” file
- Copy and paste
“HAL_GPIO_EXTI_Callback” function
from CCM lab code.txt to the start of
line 258



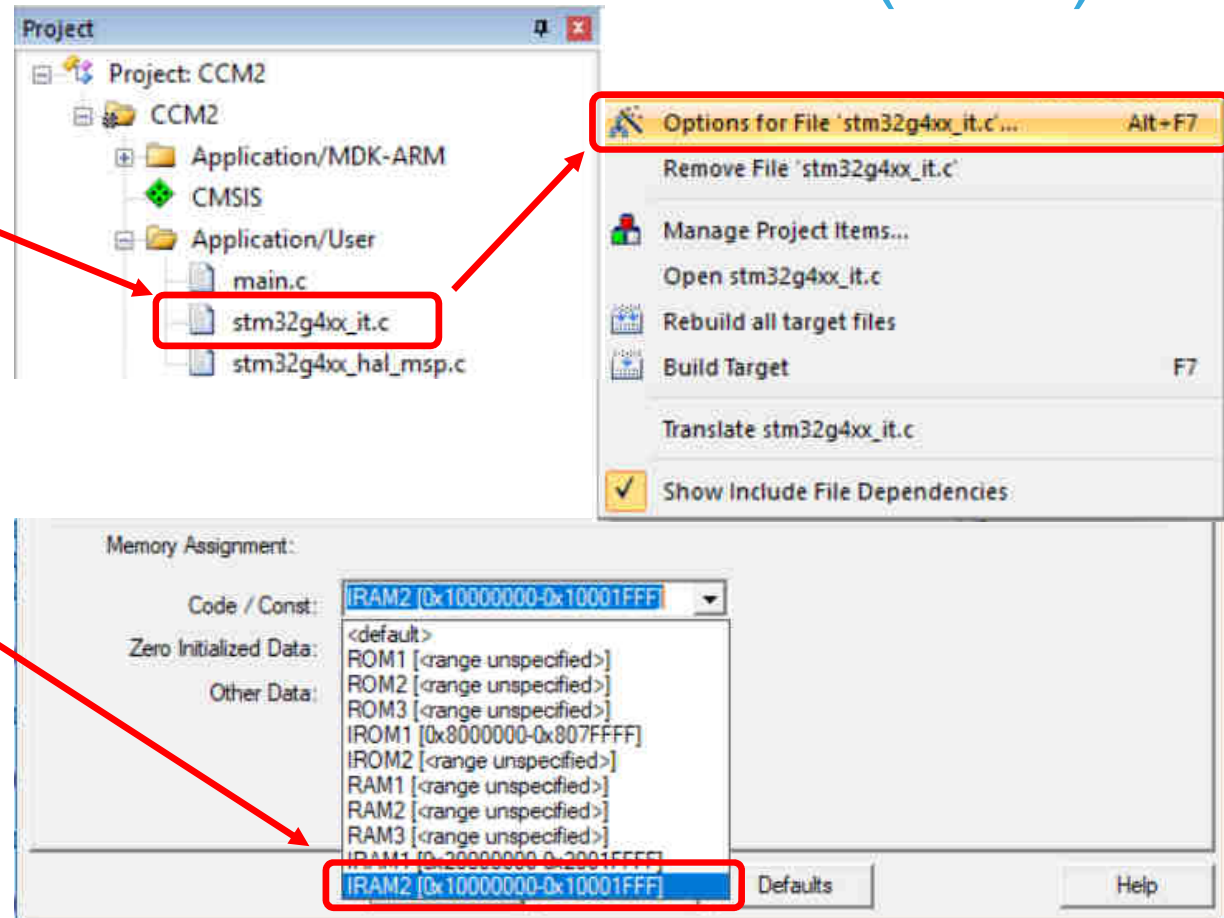
```
257  /* USER CODE BEGIN 1 */
258  void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
259  {
260      if ( GPIO_Pin == JOYSTICK_SEL_Pin )
261      {
262          HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_RESET);
263          HAL_GPIO_WritePin(LD3_GPIO_Port, LD3_Pin, GPIO_PIN_RESET);
```



CCM SRAM (files)

370

- Right mouse click on “stm32g4xx_it.c”
- Select “Options for File ‘stm32g4xx_it.c’...”
- Set Memory Assignment to “IRAM2”
- Click “OK”



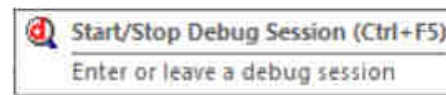
CCM SRAM (files)

371

- “Build” project



- “Start Debug” Session



- Place breakpoint in “HAL_GPIO_EXTI_Callback” (Line 260)

```
214 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
215 {
216     if ( GPIO_Pin == JOYSTICK_SEL_Pin )
217     {
```

- “Run” code



- Press Joystick button (Up, Down, Left, Right, Select)
- Breakpoint stops in CCM SRAM region address 0x10000058

```
266: }
→ 0x10000058 F5B05F00 CMP r0, #0x2000
259: {
```



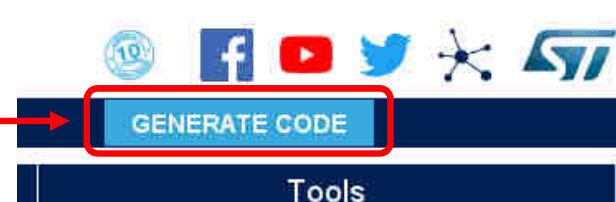
CCM SRAM (library modules)

372

- Hands-On: Place library module in CCM SRAM
- Open the CCM3.ioc CubeMx project (double mouse click)
 - Labs\CCM\CCM3\CCM3.ioc

MX CCM3.ioc

- Generate Code



- Open Project

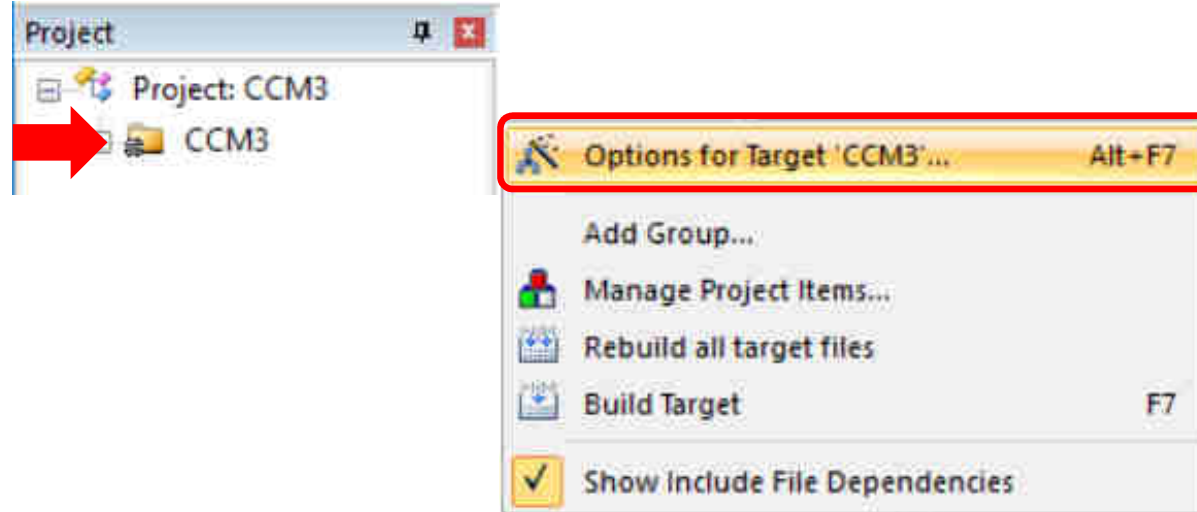




CCM SRAM (library modules)

373

- Define the memory region for the CCM SRAM in the MDK-ARM
 - Right mouse click on CCM3
 - Select “Options for Target ‘CCM3’...”

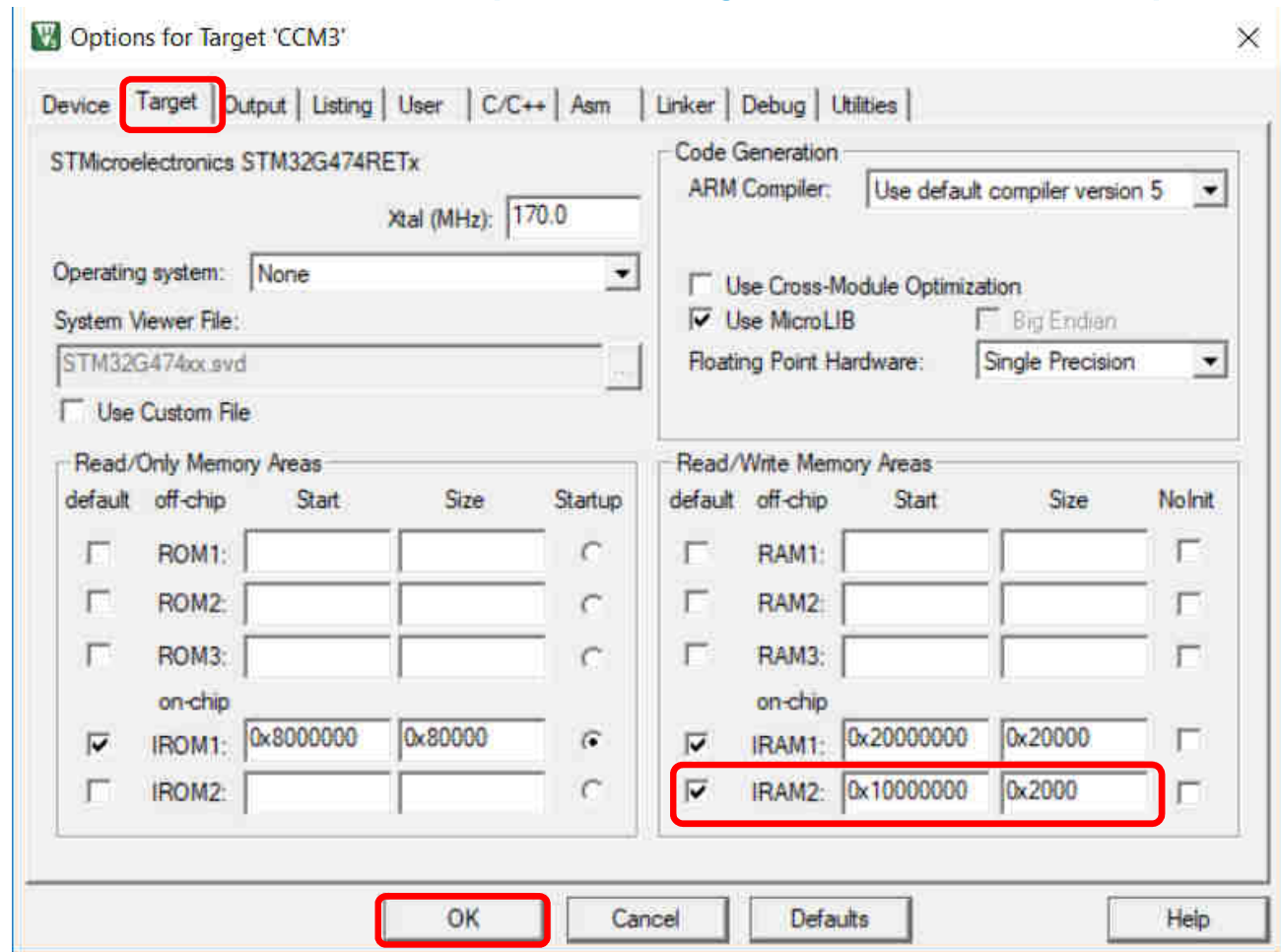




CCM SRAM (library modules)

374

- Select “Target” tab
- Check “IRAM2”
- Enter “0x10000000”
IRAM2 Start
- Enter “0x2000”
IRAM2 Size
- Click “OK”

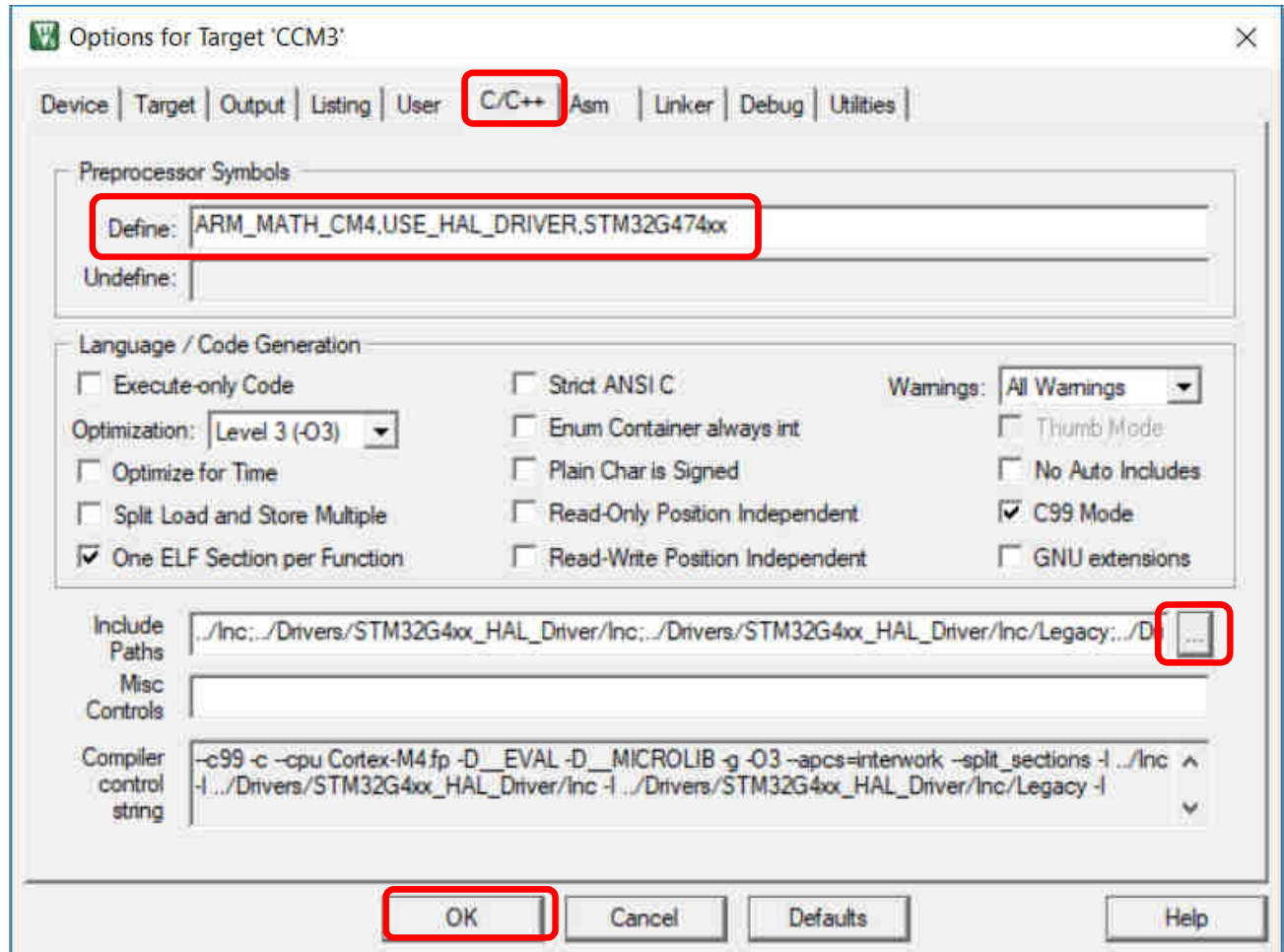




CCM SRAM (library modules)

375

- Select “C/C++” tab
- Add Define
“ARM_MATH_CM4”
- Add library include path
- Click “OK”

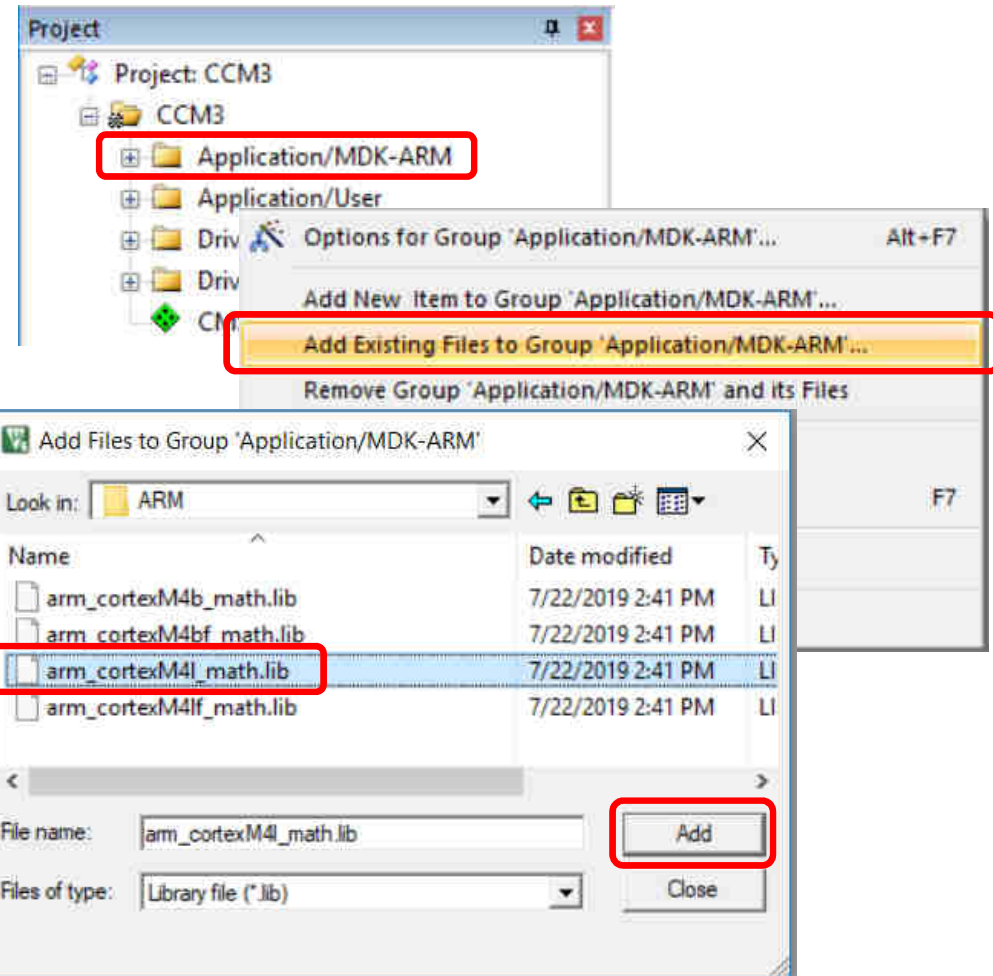




CCM SRAM (library modules)

376

- Add ARM math library
 - Right mouse click on Application/MDK-ARM
 - Add Existing Files to Group
 - Select “arm_cortexM4l_math.lib” file
 - \Drivers\CMSIS\Lib\ARM
- Click “Add” button
- “Close” window

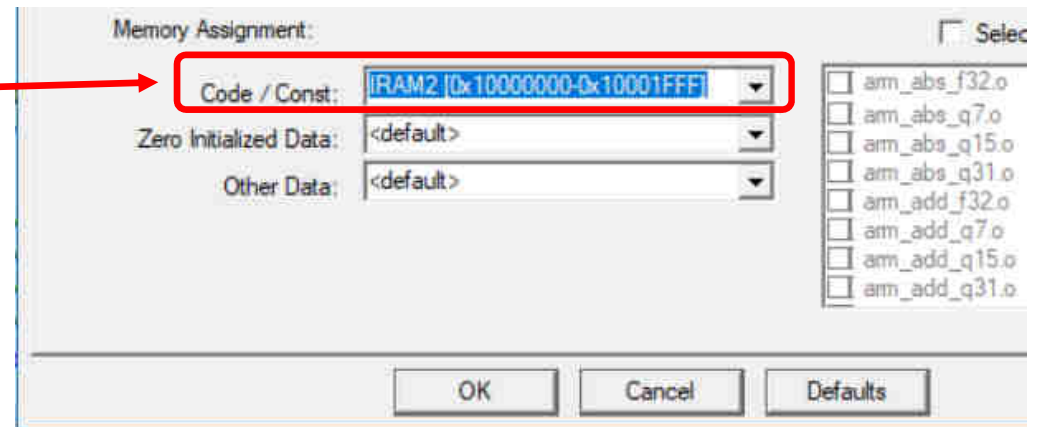
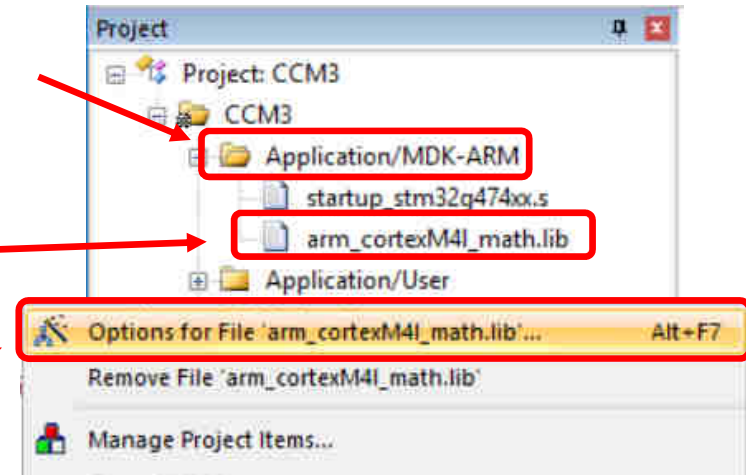




CCM SRAM (library modules)

377

- Expand “Application/MDK-ARM” folder
- Right mouse click on arm_cortexM4I_math.lib file
- Select “Options for File ‘arm_cortexM4I_math.lib’...”
- Set Memory Assignment
 - Code = IRAM2 [0x10000000-0x10001FFF]
- Click “OK”

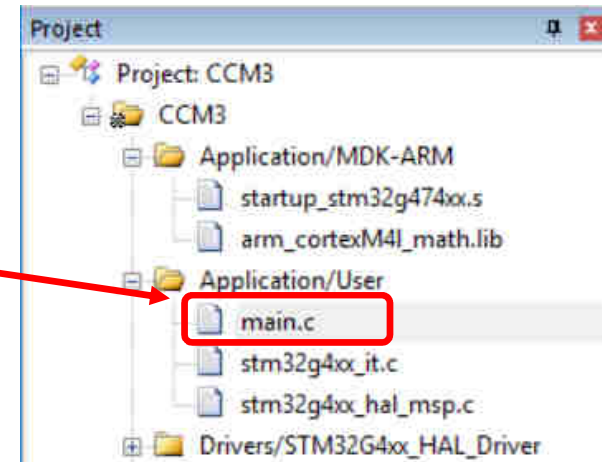




CCM SRAM (library modules)

378

- Expand “Application/User” folder
- Open “main.c” file
- Go to line 26:
 - Add: `#include <arm_math.h>`
- Go to line 36:
 - Add: `#define ARRAY_SIZE 64U`



```
24  /* Private includes -----  
25  /* USER CODE BEGIN Includes */  
26  #include <arm_math.h>  
27  /* USER CODE END Includes */
```

```
34  /* Private define -----  
35  /* USER CODE BEGIN PD */  
36  #define ARRAY_SIZE 64U  
37  /* USER CODE END PD */
```



CCM SRAM (library modules)

379

- Go to line 47:
 - Add: int32_t aAnglesLib array
 - Add: int32_t aResults array
 - Copy from CCM lab code.txt

```
46  /* USER CODE BEGIN PV */
47  /* Array of angles for arm math library Q1.31 format
48  static int32_t aAnglesLib[ARRAY_SIZE] =
49  {
50      0x00000000, 0x02000000, 0x04000000, 0x06000000,
51      0x08000000, 0x0A000000, 0x0C000000, 0x0E000000,
52      0x10000000, 0x12000000, 0x14000000, 0x16000000,
53      0x18000000, 0x1A000000, 0x1C000000, 0x1E000000,
54      0x20000000, 0x22000000, 0x24000000, 0x26000000,
55      0x28000000, 0x2A000000, 0x2C000000, 0x2E000000,
56      0x30000000, 0x32000000, 0x34000000, 0x36000000,
57      0x38000000, 0x3A000000, 0x3C000000, 0x3E000000,
58      0x40000000, 0x42000000, 0x44000000, 0x46000000,
59      0x48000000, 0x4A000000, 0x4C000000, 0x4E000000,
60      0x50000000, 0x52000000, 0x54000000, 0x56000000,
61      0x58000000, 0x5A000000, 0x5C000000, 0x5E000000,
62      0x60000000, 0x62000000, 0x64000000, 0x66000000,
63      0x68000000, 0x6A000000, 0x6C000000, 0x6E000000,
64      0x70000000, 0x72000000, 0x74000000, 0x76000000,
65      0x78000000, 0x7A000000, 0x7C000000, 0x7E000000
66  };
67
68  static int32_t aResults[ARRAY_SIZE];
```



CCM SRAM (library modules)

380

- Go to line 114:
 - Add: for loop to calculate "arm_sin_q31"
 - Copy from CCM lab code.txt



```
113  /* USER CODE BEGIN 2 */
114  for (uint32_t j = 0; j < ARRAY_SIZE; j++)
115  {
116      /* Calculate sine */
117      aResults[j] = arm_sin_q31(aAnglesLib[j]);
118  }
119
120  /* USER CODE END 2 */
```

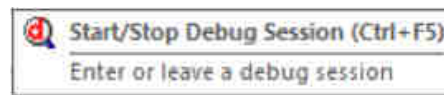
CCM SRAM (library modules)

381

- “Build” project



- “Start Debug” Session



- Place breakpoint on call to “arm_sin_q31” (Line 117)

```
114   for (uint32_t j = 0; j < ARRAY_SIZE; j++)
115   {
116       /* Calculate sine */
117       aResults[j] = arm_sin_q31(aAnglesLib[j]);
118   }
```

- “Run” code



- “Step” the code twice



- Stops in CCM SRAM region address 0x10000000

```
arm_sin_q31:
0x10000000 B470 PUSH    {r4-r6}
0x10000002 0D81 LSRS    r1,r0,#22
```



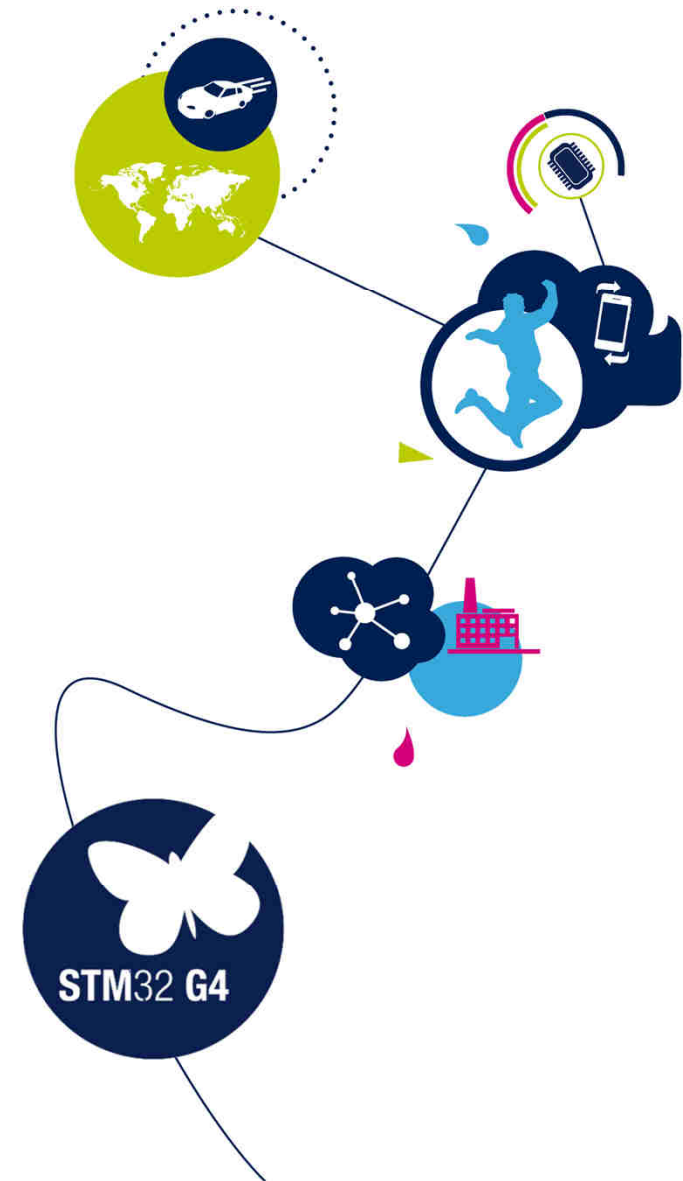
CCM SRAM: Additional Notes

382

- AN4296: Use STM32F3/STM32G4 CCM SRAM with IAR EWARM, Keil MDK-ARM and GNU-based toolchains

STM32G4 mixed-signal MCU workshop

Wrap Up



STM32G4 Workshop Related Survey

384

Use your phone to scan the QR code or type the link into your browser.



<https://www.surveymonkey.com/r/PC5DHBM>



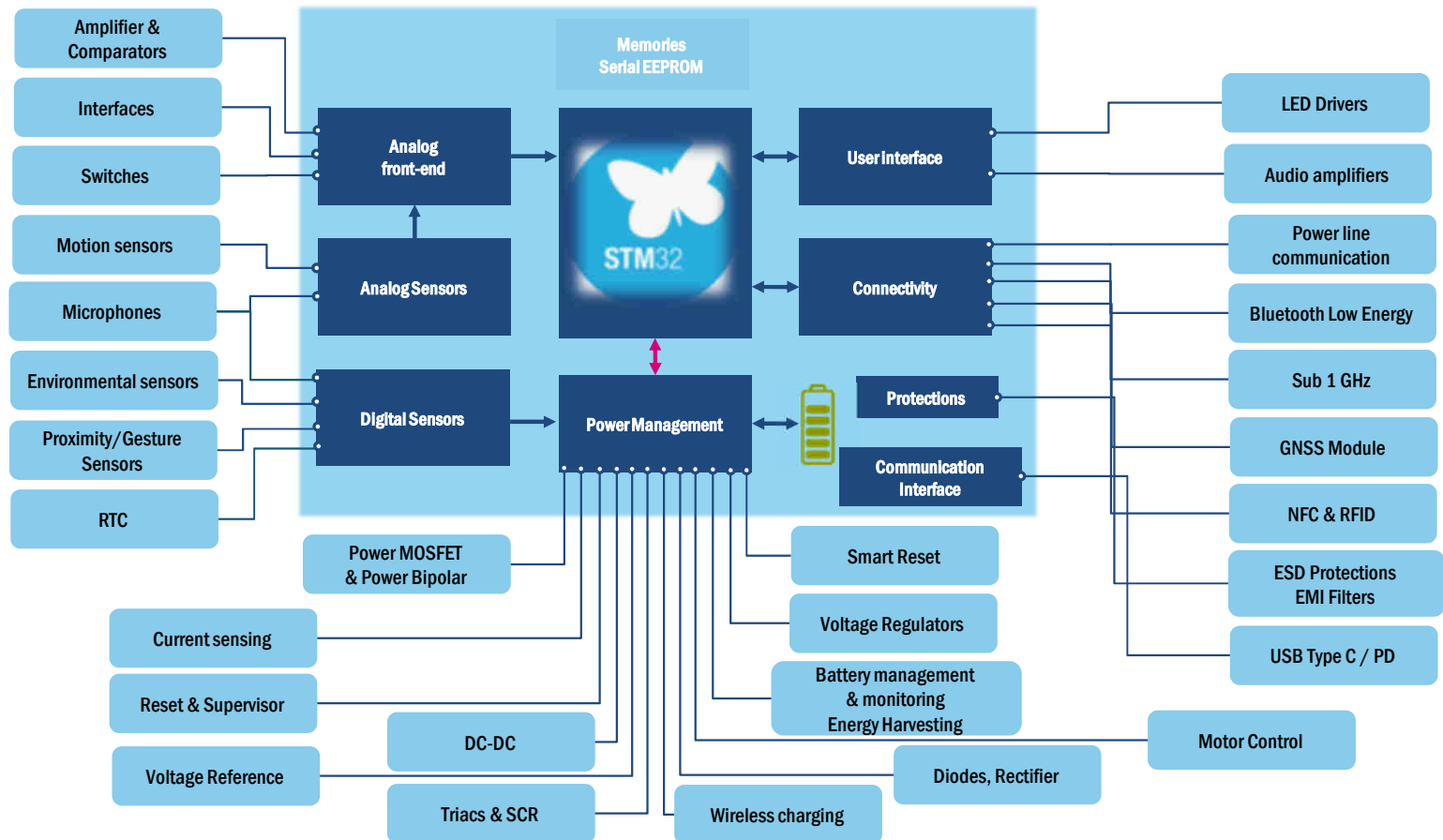
We Greatly Value Your Feedback - Thank you!

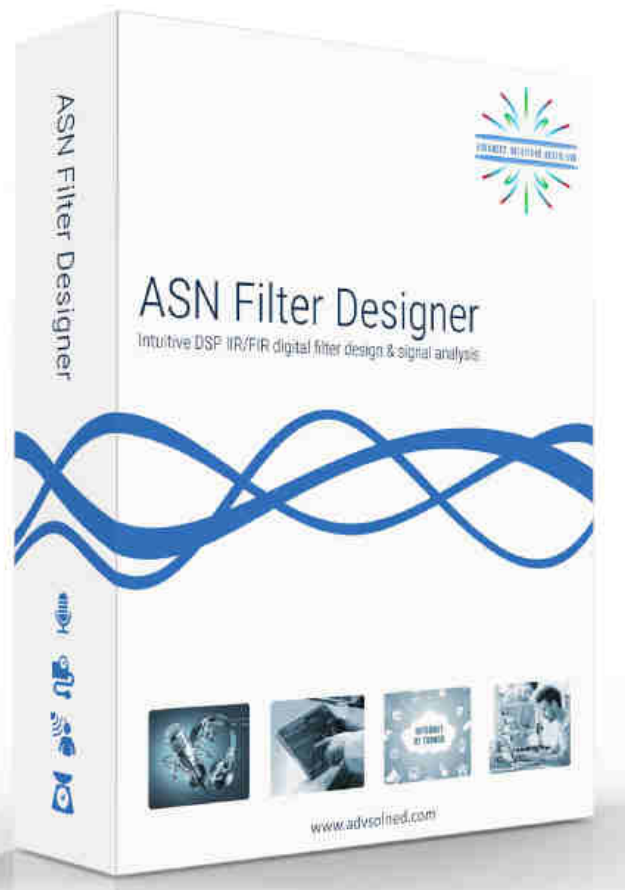
STMicroelectronics: one of the broadest Standard Product Portfolios

The largest offerings of companion products for every **STM32** Microcontrollers



- Protection devices
- Power bipolar transistors
- Power MOSFETs
- Amplifiers and Comparators
- Linear voltage regulators
- EEPROM
- Rectifiers
- SCRs and TRIACs
- Interface ICs
- Voltage references
- Reset and supervisor ICs





ASN Filter Designer

386

Powerful DSP experimentation platform

Design and fine-tune your digital filter within minutes, and then test it in real-time with a variety of test signals (including your own datasets) via the tool's built-in signal analyser.

Real-time design feedback

No need to explicitly define technical specifications, just sketch your requirements and let the tool automatically fill in the exact specifications for you!

Live math scripting

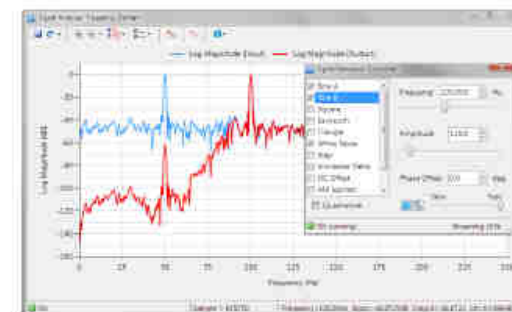
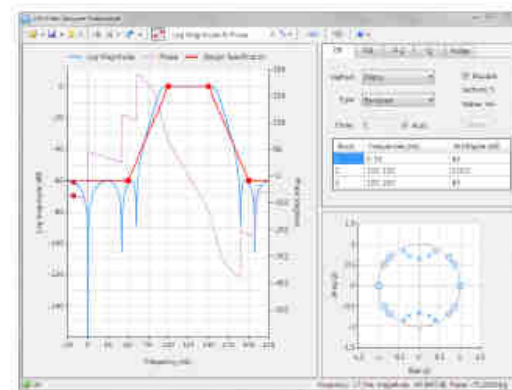
Interactively experiment with specialised transfer functions, pole-zero pairs and other mathematical functions with ASN FilterScript.

Automatic documentation

Get a detailed specification report of all designed filter blocks for official project documentation, which can be pasted directly into Microsoft Word/Powerpoint.

Deploy to Arm CMSIS-DSP, C/C++, Python, Matlab and more

Easily integrate your designed filter blocks with your other algorithms in Arm CMSIS-DSP, C/C++, Python, Matlab, Octave, Scilab or Labview using provided royalty free software development frameworks.



Download a free demo

http://www.advsolned.com/asn_filter_designer/



Biricha Digital Power Design Tool

387



Digital Power workshops with BIRICHA

381

How to implement Digital Power Supplies (PSU) and Power Factor Correction (PFC)

STM32 PSU/PFC design
Step-in Digital Power technology



Biricha Digital Power:
World leading expertise and training in Digital Power

Workshop based on STM32F334 and STM32G474
Nucleo and its dedicated Digital Power expansion
board

STMicroelectronics brings industry leading ST MCUs
together with Biricha's tools and training





Appendix – Reference Documents

388

Datasheet and Reference Manual Links

Datasheets

<u>STM32G431xx</u>	STM32G431x6 STM32G431x8 STM32G431xB
<u>STM32G441xB</u>	STM32G441xB
<u>STM32G473xx</u>	STM32G473xB STM32G473xC STM32G473xE
<u>STM32G474xx</u>	STM32G474xB STM32G474xC STM32G474xE
<u>STM32G483xE</u>	STM32G483xE
<u>STM32G484xE</u>	STM32G484xE

Reference Manual

<u>RM0440</u>	STM32G4 Series Reference Manual
-------------------------------	---------------------------------



Appendix – Reference Documents

389

Application Note Links

<u>AN2834</u>	How to get the best ADC accuracy
<u>AN4013</u>	STM32 Cross series timer overview
<u>AN4232</u>	Getting started with analog comparators
<u>AN4296</u>	Using CCM SRAM with IAR, Keil, and GNU based tool chains
<u>AN4539</u>	HRTIM Cookbook
<u>AN4899</u>	STM32 GPIO configuration for hardware settings and low power consumption
<u>AN4989</u>	STM32 debug toolbox
<u>AN5305</u>	Digital filter implementation with the FMAC
<u>AN5306</u>	Operational Amplifier usage in STM32G4
<u>AN5310</u>	Guideline for using analog features of the G4 vs. the F3
<u>AN5315</u>	STM32Cube firmware examples for STM32G4
<u>AN5325</u>	Getting started with the CORDIC accelerator
<u>AN5345</u>	High-brightness RGB LED control using the B-G474E-DPOW1 Discovery kit
<u>AN5346</u>	STM32G4 use tips and recommendations



Appendix – Reference Documents

390

User Manual Links

<u>UM1721</u>	Developing applications on STM32Cube with FatFS
<u>UM1722</u>	Developing applications on STM32Cube with RTOS
<u>UM2454</u>	STM32G4 Series Safety Manual
<u>UM2492</u>	Getting started with STM32CubeG4 for STM32G4 Series
<u>UM2552</u>	USB Power Delivery
<u>UM2570</u>	Description of STM32G4 HAL and low-layer drivers
<u>UM2573</u>	STM32CubeG4 Nucleo demonstration firmware
<u>UM2577</u>	Discovery kit with STM32G474RE MCU



STM32G4 Series – Take Away

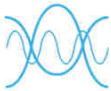
391

Analog-rich MCUs for mixed-signal applications



Performance

170MHz Cortex-M4 coupled with 3x accelerators

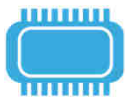


Rich and Advanced Integrated Analog

ADC, DAC, Op-Amp, Comp.



Safety and security focus



Large portfolio available from NOW!

32..512KB Flash memory

32..128-pin packages

Releasing Your Creativity

392

